CMP
United Business Media

PRODUCT REVIEWS: HAVOK 2, FACEGEN MODELLER 2.2, MORE

JUNE 2003

GameDeveloper

ON THE FRONT LINE OF GAME INNOVATION

Design Feedback That Won't Leave You Hanging

Usability Testing for Games

Beautiful, Yet Efficient Performance-Friendly Content Creation

Developing the Will to Kill: The Real Story Behind RATCHET & CLANK

$5.95US $6.95CAN
06
71486 02133 9
Display Until June 27, 2003

YOUR VILLAINS
YOUR BRUTES
YOUR ASSASSINS
YOUR OGRES
YOUR HEROINES
OUR PACKAGE

OUR STRENGTHS AT YOUR SERVICE
OUR GRAPHIC AND STRUCTURAL DESIGNS
OUR KNOWLEDGE OF MATERIALS: PAPERBOARD, PLASTIC, SPECIAL PAPERS AND FOILS, METALS
OUR KNOWLEDGE OF FINISHING: STAMPING, EMBOSSING, LAMINATING
OUR CONSISTENCY OF PERFORMANCE
OUR INTERNATIONAL CAPABILITY
OUR LONG TERM PRESENCE IN ENTERTAINMENT PACKAGING
YOUR EXPECTATIONS FULLY JUSTIFIED

CALL AGI NOW TO TAKE ADVANTAGE OF OUR STRENGTHS IN YOUR BUSINESS

AGI Media
A MeadWestvaco Resource

Contact: Lee Ramsey 317.241.1745 www.agimedia.com

CONTENTS

VOLUME 10, NUMBER 6

June 2003

FEATURES

34 Improving Games with User Testing: Getting Better Data Earlier

How can game teams get design feedback earlier, when the game isn't in final form yet? Several possible methods come from a discipline called usability. Central to the concept of usability is the evaluation of users, or target audience. As Federoff digs deeper into the concept, she provides solutions that just might be the right one for your company.

Melissa Federoff

42 Beautiful, Yet Friendly, Part 1: Stop Hitting the Bottleneck

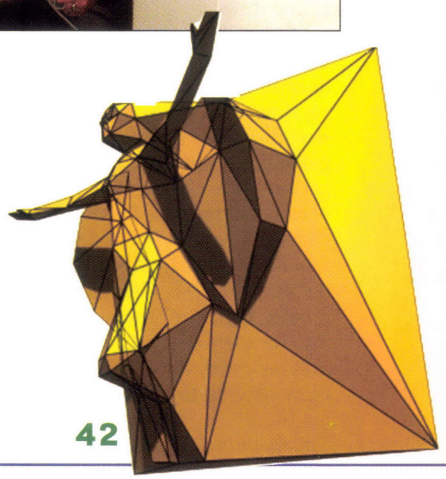
A programmer's job isn't to make a good-looking game, it's to empower the artists to make a good-looking game. In this two-part article on graphics performance, Provost explains how to manage performance without losing track of this vision statement.

Guillaume Provost

52 Postmortem: Insomniac's Ratchet & Clank

It was late spring 2000 and the guys at Insomniac knew they had to start planning for their first PS2 project. Their problem, however, was twofold: they had decided not to develop any more Spyro games, and they were deciding whether they wanted to stay with the platform-action genre. Price takes us down a road that, while being wide open, presents the challenge of figuring out which direction to take.

Ted Price



DEPARTMENTS

4 **GAME PLAN** *Jennifer Olsen*
Credits and Debts

6 **INDUSTRY WATCH** *Everard Strong*
Nintendo cuts royalties, Xbox cuts prices, Maya 5 announced, more

8 **PRODUCT REVIEWS**
Havok 2, FaceGen Modeller 2.2, *Real-Time Shader Programming*

14 **PROFILES** *Everard Strong*
Monolith's Samantha Ryan on team leadership

COLUMNS

18 **THE INNER PRODUCT** *Jonathan Blow*
Unified Rendering LOD, Part 4

24 **ARTIST'S VIEW** *Hayden Duvall*
Transition

30 **SOUND PRINCIPLES** *Mike Verrette*
Managing the Nonlinear Mix

32 **BETTER BY DESIGN** *Noah Falstein*
The Hobgoblin of Little Minds

56 **SOAPBOX** *Ben Sawyer*
Gaming Our Way to Our Better Future

COVER: *The cover image was created through a combination of Maya, Photoshop, and Insomniac's screenshot utilities. A high-resolution image of Ratchet's real-time game model was posed and rendered in Maya. Next, an environment from the game was found and captured as a high-resolution screenshot. The image then spent a few days in Photoshop, where touch-ups and additional details were painted in. Thanks to the Insomniac art team, and to Greg Baldwin and David Guertin in particular.*

Game Developer (ISSN 1073-922X) is published monthly by CMP Media LLC, 600 Harrison St., San Francisco, CA 94107, (415) 947-6000. Please direct advertising and editorial inquiries to this address. Canadian Registered for GST as CMP Media LLC, GST No. R13288078, Customer No. 2116057, Agreement No. 40011901. **Subscription Rates:** Subscription rate for the U.S. is $49.95 for twelve issues. Countries outside the U.S. must be prepaid in U.S. funds drawn on a U.S. bank or via credit card. Canada/Mexico: $69.95; all other countries: $99.95 (issues shipped via air delivery). Periodical postage paid at San Francisco, CA and additional mailing offices. **Postmaster:** Send address changes to Game Developer, P.O. Box 1274, Skokie, IL 60076-8274. **Customer Service:** For subscription orders and changes of address, call toll-free in the U.S. (800) 250-2429 or fax (847) 647-5972. All other countries call (1) (847) 647-5928 or fax (1) (847) 647-5972. Send payments to *Game Developer*, P.O. Box 1274, Skokie, IL 60076-8274. For back issues write to *Game Developer*, 1601 W. 23rd St. Ste. 200, Lawrence, KS 66046-2703. Call toll-free in the U.S./Canada (800) 444-4881 or fax (785) 841-2624. All other countries call (1) (785) 841-1631 or fax (1) (785) 841-2624. Please remember to indicate *Game Developer* on any correspondence.

Swinging For The Fences

Taking Sega Sports™ World Series Baseball® 2K3 from XBox™ to PlayStation®2

daniel r. huebner

From the start, World Series Baseball 2K3 looked to be a challenging project. As with any sports game, WSB required high frame rates, fast response to player input, and a huge number of animations to support fast-paced action. Recreating the baseball experience also meant implementing statistically realistic gameplay and rendering as many as 50 high-resolution characters at one time.

Blue Shift set out in May 2002 with a disarmingly simple mandate: every version of World Series Baseball 2K3 would achieve the same level of gameplay, appearance, and performance. The game had originally been developed for Dreamcast, but that console's demise led to a port and rebuild for Xbox, which would in turn be ported to PS2. Creating a high profile, high performance AAA sports title for the major consoles requires milking every bit of performance the individual hardware has to offer.

The team understood from the beginning that targeting both platforms simultaneously in such a short development cycle would be a taxing endeavor. Blue Shift had experience with Xbox, but this project would be their first chance to develop for PS2. They knew that reaching their goal of equal performance across platforms would require an extremely ambitious technology plan. As Blue Shift had just nine months to bring this installment of World Series Baseball from Xbox to PS2, developing completely new engine technology simply wasn't an option.

Buy, Build, or Both?

The decision was made early on to use a third-party rendering engine that offered the flexibility to aggressively target each platform with a combination of custom rendering pipelines and newly written subsystems. World Series Baseball 2K3 was developed by a large, distributed team working on both platforms simultaneously, and the short development cycle meant that any custom engine technology would have to be developed in parallel with gameplay and art assets. As rendering engine components or plug-ins became available, these new components would be integrated into the game. In order to minimize potential communication and synchronization problems, Blue Shift wanted to start with a standard, well documented API and a common toolset for multi-platform development that would allow both programmers and artists to get up to speed quickly without having to be overly concerned with console-specific issues. RenderWare offered Blue Shift robust multi-platform performance that they could quickly customize to meet their goals.

BLUE SHIFT INCORPORATED

Blue Shift Engine Technologist Alex Pepper

"We realized that it was too expensive and impractical to completely build a new engine from the ground up..."

The decision wasn't based solely on expediency; Blue Shift's first swing at World Series Baseball was a version for Dreamcast that saw the studio creating an engine from the ground up only to see the game ship after the Dreamcast itself had been canned. This experience convinced the company that custom technology wasn't necessarily the best way to go. "We realized that it was too expensive and impractical to completely build a new engine from the ground up for every new console that was released," explains Blue Shift Engine Technologist Alex Pepper, "We decided that it would be useful to use a third party engine with a good tool chain and documentation that we could leverage, while still giving us the flexibility to make high performance modifications."

High Performance

Blue Shift had the daunting task of building the game they envisioned on Xbox and then bringing it intact to the more memory constrained PS2. To achieve the level of performance Blue Shift was after on each individual platform, the team chose to build on the standard RenderWare rendering pipelines with highly tuned custom pipelines created specifically for each console. The developers used RenderWare as a jumping off point on each platform, using RenderWare's structure as a template to create pipelines more precisely tuned for World Series Baseball. "The standard RenderWare pipelines helped us get started on each new platform," explains Pepper, "then we improved performance as each new pipeline came online."

Profiling and tuning were key components in Blue Shift's drive to reach their performance goals. During the development of World Series Baseball 2K3, the team created new profilers dedicated to each platform to seek out hotspots or performance bottlenecks. On Xbox they created a profiler that runs on the PC while collecting real time information from the Xbox to provide visual information on CPU and GPU timing and timing dependencies. For PS2 the team implemented a combination of fine and coarse grain profilers to first find bottlenecks at a high level then drill

down and determine key optimization areas. Once problem areas had been identified, the team optimized or replaced any sections of code to remove or minimize performance bottlenecks.

Texture Compression

Memory constraints were the prime concern when Blue Shift set out to take World Series Baseball from Xbox to PS2. While the team had achieved satisfactory results with standard DXT compression on Xbox, meeting their mandate of equal performance on every platform - along with a desire to use the same art assets across platforms - pushed them to develop a proprietary texture compression system for PS2.

Decompression of textures on the PS2's Graphics Synthesizer allowed textures to be stored in the main memory and uploaded to VRAM in a compressed format. Close integration with Blue Shift's custom rendering pipelines provided per-polygon mipmap selection and dynamic tracking of texture decompression to reduce texture upload and decompression overhead. According to Pepper, the custom implementation offers compression up to three times greater than that available with DXT 1 and allowed the team to fit the equivalent of 17MB of 16 bit textures in just 2MB of PS2 VRAM, all of which helped Blue Shift achieve the same level of texture detail on both Xbox and PS2.

to its limits, and they took the opportunity RenderWare offers to quickly implement new multi-platform systems as well. A real-time player shadowing system originally developed for the Xbox version was ported to PS2 in just one week, including additional performance and memory improvements. Blue Shift also added a new particle system engine and tools on top of RenderWare's to create dust and firework effects across platforms. An aggressive multi-platform animation compression system was put in place to handle the large number of player animations, and a multi-platform audio system rounded out Blue Shift's ambitious technology development plans.

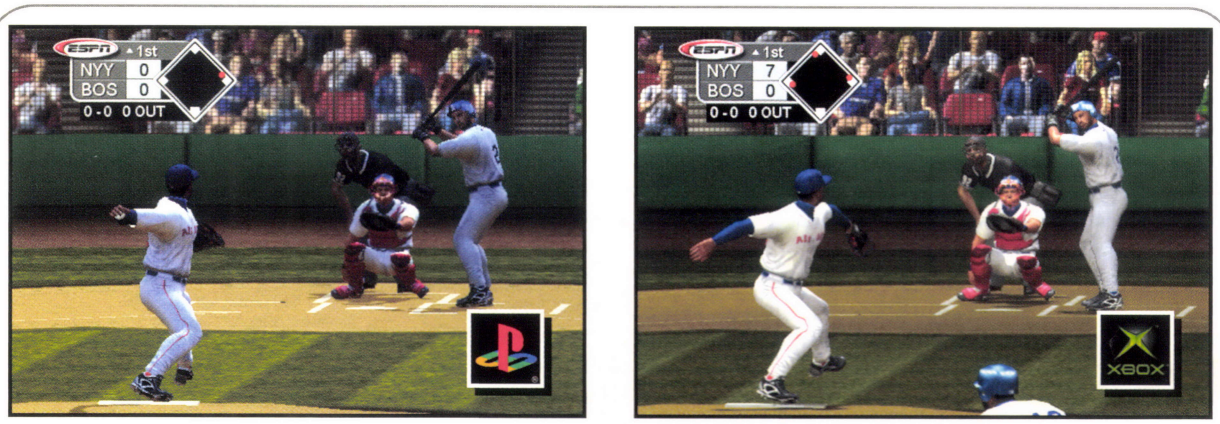


Fig 1 - RenderWare allowed Blue Shift to achieve phenomenal graphics on all platforms

"In its PS2 debut, WSB 2K3 is a game to beat." Gamepro.com

"Last year, I called WSB the Mona Lisa of video baseball. This time around, it's even better." Xbox Nation Magazine

"WSB 2K3 is the best console baseball game I've ever played." IGN.com

"Our texture compression was implemented using the RenderWare plug-in and toolkit framework for extending the base behavior of the texture object," said Pepper, " We also made changes to the texture cache system to improve upload performance for our compressed textures."

Blue Shift created a RenderWare texture plug-in to add additional compressed image data to the standard RwTexture object and a texture dictionary tool to convert textures into a platform-specific format able to be streamed at game load time using custom streaming functions built into Blue Shift's texture plug-in. Texture upload functions interface to a largely stock RenderWare texture cache to manage compressed texture VRAM uploads, with some functions added to reduce overhead.

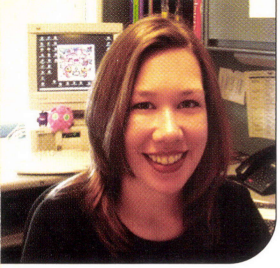
To smooth the path from art to engine, the team created standalone compression tools and dedicated Photoshop plug-ins. Artists are able to select compression formats based on size, speed and quality tradeoffs. These settings are then stored in a configuration file so that textures can be reprocessed from originals using a batch process. Texture settings can be selected for all platforms within the same interface and previewed inside Photoshop.

Multi-Platform

Not all of the technologies Blue Shift built onto the RenderWare framework were designed to address console-specific performance issues. Blue Shift is a technology intensive company that strives to push hardware

Flexibility

Bringing World Series Baseball 2K3 to Xbox and PS2 in just nine months required a huge amount of design and technological flexibility. Blue Shift wanted the advantages of a robust, stable multi-platform technology platform, but required the flexibility to build out custom systems to meet an aggressive technology plan and performance goals. By concentrating time and resources on the systems and features that could radically improve performance, Blue Shift was able to meet their self-imposed mandate: every version of World Series Baseball 2K3 achieved very similar levels of gameplay, appearance, and performance.



GAME PLAN

LETTER FROM THE EDITOR

Credits and Debts

In an industry where days are long but memories are often short, game credits are vital, but they sometimes pose problems for developers and employers. If you've never had to confront the issue head-on in your professional career, consider yourself lucky. When a former employer owns every last byte of your work on a game, your name in the credits may be all you have to show for your work and help land your next job. There are opportunities for abuse on the parts of both employer (denial of credit) and employee (credit inflation or misrepresentation); fortunately, there are incentives for both sides to resolve the issue.

One stumbling block is the lack of industry-wide standardization of job titles and their respective scope of responsibility, which creates opportunity for abuse from both employers and employees. In Hollywood, people work on contracts and deal with unions that can specify what a "best boy" is and does, so it's hard for him to show up on his next project claiming he was a second-unit director. On the flip side, the union would be able to protect him through arbitration if he fulfilled the terms of his contract but the film's fun-loving producers decided to credit him as "goat boy" instead.

More significant than the mere lack of job-title standardization in the game industry is the anarchy that characterizes the methodology around which credit files are typically created, maintained, and implemented at the end of a project. Many employers still have a make-it-up-as-they-go-along approach to generating game credits, both because there is no industry-standard methodology to follow and because disputes often arise after the fact and are forgotten by the time the next title ships. The IGDA is looking at creating a voluntary set of standards for companies to follow, which would be of great help to employers and great relief to developers.

The use of short-term contractors and mid-project turnover of full-time employees represent two big sticking points in credit determination. As lengths of game projects swell from "Gee, I could have gotten a master's instead" to "I gave you the best years of my life!" the chances of an entire team remaining intact from start to finish are increasingly slim. Since most work in the game industry is done by studio employees rather than contractors, mid-project departures wreak havoc with credit claims. One company might punish a lead who left a month before ship by eliminating him or her from the credits, while another might include a low-level person who got fired for incompetence after a few months. It's a crapshoot for hardworking developers who don't know whether they are choosing between their next opportunity and credit for their current work.

What can you do to protect yourself? First, make it a priority to protect the investment your hard work on a project represents. This means that you can't rely on your employer's good intentions. Raise the subject of credits with the project or studio manager when you are hired, or each time you start on a new project. Find out who administers the credit file and how it is reviewed. There should be a consistent framework to determining criteria for inclusion (and exclusion), and you should get it in writing. Such a framework benefits employees by managing their expectations and employers by giving them the ability to justify and defend their final decisions.

Game developers are not by nature a vainglorious lot; most are still doing it out of love for the work rather than a shot at notoriety. The idea of pushing for verifiable crediting processes at your company may even feel unseemly or selfish. But credits are an area where there should never be surprises. It's worth it to both you and your employer to establish what your company's crediting standards are.

Jennifer

Jennifer Olsen
Editor-In-Chief

GameDeveloper
www.gdmag.com
600 Harrison Street, San Francisco, CA 94107 t: 415.947.6000 f: 415.947.6090

Publisher
Jennifer Pahlka *jpahlka@cmp.com*

EDITORIAL
Editor-In-Chief
Jennifer Olsen *jolsen@cmp.com*
Managing Editor
Everard Strong *estrong@cmp.com*
Production Editor
Olga Zundel *ozundel@cmp.com*
Art Director
Audrey Welch *awelch@cmp.com*
Editor-At-Large
Chris Hecker *checker@d6.com*
Contributing Editors
Jonathan Blow *jon@number-none.com*
Hayden Duvall *haydend@3drealms.com*
Noah Falstein *noah@theinspiracy.com*
Advisory Board
Hal Barwood LucasArts
Ellen Guon Beeman Monolith
Andy Gavin Naughty Dog
Joby Otero Luxoflux
Dave Pottinger Ensemble Studios
George Sanger Big Fat Inc.
Harvey Smith Ion Storm
Paul Steed Microsoft

ADVERTISING SALES
Director of Sales/Associate Publisher
Michele Sweeney e: *msweeney@cmp.com* t: 415.947.6217
Senior Account Manager, Eastern Region & Europe
Afton Thatcher e: *athatcher@cmp.com* t: 828.350.9392
Account Manager, Northern California & Southeast
Susan Kirby e: *skirby@cmp.com* t: 415.947.6226
Account Manager, Recruitment
Raelene Maiben e: *rmaiben@cmp.com* t: 415.947.6225
Account Manager, Western Region & Asia
Craig Perreault e: *cperreault@cmp.com* t: 415.947.6223
Account Representative
Aaron Murawski e: *amurawski@cmp.com* t: 415.947.6227

ADVERTISING PRODUCTION
Vice President, Manufacturing Bill Amstutz
Advertising Production Coordinator Kevin Chanel
Reprints Terry Wilmot e: *twilmot@cmp.com* t: 516.562.7081

GAMA NETWORK MARKETING
Director of Marketing Greg Kerwin
Senior MarCom Manager Jennifer McLean
Marketing Coordinator Scott Lyon

CIRCULATION
BPA INTERNATIONAL
Game Developer is BPA approved
Group Circulation Director Catherine Flynn
Circulation Manager Ron Escobar
Circulation Assistant Ian Hay
Newsstand Analyst Pam Santoro

SUBSCRIPTION SERVICES
For information, order questions, and address changes
t: 800.250.2429 or 847.647.5928 f: 847.647.5972
e: *gamedeveloper@halldata.com*

INTERNATIONAL LICENSING INFORMATION
Mario Salinas
t: 650.513.4234 f: 650.513.4482 e: *msalinas@cmp.com*

CMP MEDIA MANAGEMENT
President & CEO Gary Marshall
Executive Vice President & CFO John Day
Chief Operating Officer Steve Weitzner
Chief Information Officer Mike Mikos
President, Technology Solutions Group Robert Faletra
President, Healthcare Group Vicki Masseria
President, Electronics Group Jeff Patterson
Senior Vice President, Global Sales & Marketing Bill Howard
Senior Vice President, HR & Communications Leah Landro
Vice President & General Counsel Sandra Grayson
Vice President, Creative Technologies Philip Chapnick

CMP
United Business Media

GamaNetwork

WWW.GAMANETWORK.COM

BUILD A GREAT GAME, WIN $100,000!

And that's only the beginning...

Design and develop your game and enter it into the RealOne™ Arcade Game Developer Showdown.

You could win the $100,000 Grand Prize or one of two $25,000 Runner-Up prizes. With RealOne™ Arcade distribution, you own the games and reap the rewards.

SIGN UP NOW AT
http://gamedevs.realonearcade.com

ULTRA OFFICIAL CONTEST RULES

1. Contest. The RealOne Arcade Game Developer Showdown ("Contest") is sponsored by RealNetworks, Inc., located at 2601 Elliot Avenue, Seattle, Washington, 98121 ("RN"). The Contest will begin at or around 12:01 PM E.S.T. on July 1, 2003 and will end at or around 11:59 PM E.S.T. on July 31, 2003 (the "Contest Period"). Please direct all inquiries about the Contest to gamedevs@real.com. Details about the Contest, official entry forms, and these Official Rules will be available on http://gamedevs.real.com/contest/. 2. Eligibility. The Contest is open to individuals residing in the United States of America and Canada who are at least 18 years of age or older at the time of entry ("Entrants"). Entrants may also be organizations (such as corporations or limited liability partnerships) validly organized and in good standing in any United States or Canadian jurisdiction. Employees, contractors, directors, agents, representatives, and the immediate family members of RN, subsidiaries and advertising and promotion agencies are ineligible. The Contest is subject to all applicable federal, state and local laws. The Contest is VOID WHERE PROHIBITED. 3. Contest Timeline. Entrants may submit their Game (as defined below) at any time during the Contest Period. Finalists will be chosen on or about August 15, 2003. One (1) Grand Prize Winner and two (2) Runner-Up Winners will be chosen on or about August 30, 2003. 4 Entry Requirements. Entrants must create and design an original downloadable game software program that has not been distributed or released to the general public in any distribution channel ("Game"). One (1) Game per Entrant. All Games shall remain the property of the Entrant. Entries must be completed online per the instructions at http://gamedevs.real.com. The following are requirements for entry into the Contest and, except as specifically set forth herein, failure to meet any of the below requirements may immediately disqualify the Game from consideration: (a) Each Entrant must complete the entry form (available at http://gamedevs.real.com/contest/) and successfully submit the form during the Contest Period. (b) Each Entrant must submit the Game in the form and manner described in the entry form during the Contest Period. (c) Each Game must comply with the following technical requirements: (i) It must have incorporated a self-executing game installer designed to run on the Windows XP operating system. (ii) It shall require no more than 64MB RAM to run. (iii) It shall run on a Matrox Millenium G400 video card. (iv) It shall be less than 100 MB in total compressed download size. (v) All components of the Game must be owned by the Entrant or under a full license for the uses described in this (vi) All technological elements needed to play the game must be installed in the submitted Game (e.g. if it needs java, java must be installed). (vii) Game play must be fully controlled by mouse or keyboard, or mouse/keyboard combination. (viii) All game text must be in English. (ix) It must be free of any inappropriate content (pornography, racism, sexism, extreme violence, extreme profanity), as determined by RN at its sole discretion. (x) It must be free of any known technological defects (e.g. viruses, time-bombs, etc.). (xi) It must be playable without collection of any personally identifiable information by users. (xii) It may not contain advertising or links to any third party web site. (d) Each Entrant must provide applicable Game documentation that at a minimum describes how to play the Game, and may feature key aspects of the Game the Entrant would like to call attention to. (e) Each Entrant must complete and sign a one-year standard distribution agreement with RN to distribute the Game via RealOne Arcade ("Agreement") as part of the Submission process, which shall include a three (3) month window in which RN shall be the exclusive distributor of the Game. RN is not obligated to distribute any Game. (f) Additionally, it is suggested (but not mandatory) that each Game include a URL to launch an online web version of the Game for a web browser running on a Windows PC (IE 4.0 or higher preferred). (g) a Game is submitted, an Entrant may not make any further changes or modifications to the Game. 5. Judging Process. All Games will judged by a panel of six (6) RealNetworks employees ("Judges"). Each Game will be judged on the following criteria ("Criteria"): (a) ease of use; (b) smallest download size; (c) how much fun the game is to play; (d) stability of play; (e) production quality; and (f) originality. All Judges will carefully review each eligible Game based on the Criteria, complete an evaluation form, and assess a total score. All Criteria will be weighted equally. The total scores submitted by the Judges for each Game will be averaged, and the ten (10) Games with the highest average scores will become finalists. The Judges will review the finalists' Games and, based upon consensus, will determine the Grand Prize Winner and Runner-Up Winners. The Judges will determine assignment of all prizes, and all decisions by Judges regarding winners are final. 6. Prizes. One (1) "Grand Prize Winner" will receive a cash prize of one hundred thousand dollars ($100,000.00). Two (2) "Runner-Up Winners" will each receive a cash prize of twenty five thousand dollars ($25,000). All cash awards shall be awarded in U.S. Dollars. Total value of all prizes is one hundred and fifty thousand U.S. dollars ($150,000). All prizes will be awarded. The winners will be notified in writing at the physical address provided in the entry form within ten (10) days of August 30, 2003. Prize notifications returned as undeliverable or the failure of the winner to prove eligibility will result in an alternative winner being selected. Only the prizes set forth above will be awarded. Prizes are non- transferable. The tentative dates for prize fulfillment are between September 1, 2003 and October 1, 2003 and are subject to change at RN's sole discretion. RN will make arrangements for the fulfillment of all prizes. The value of the prize(s) may be taxable as income. 7. Winners. All winners must: (a) sign an affidavit of eligibility and release RN from any and all liability, claims, demands, and causes of action for personal injury and/or damage, theft, or loss suffered in connection with this Contest or the use or acceptance of the prize or any portion thereof to be eligible for the prizes ("Release"); and (b) complete and sign any applicable tax forms provided by RN ("Tax Form"). Winners must complete the Release and Tax Form and return them to RN within five (5) business days or risk disqualification and the selection of an alternate winner by RN. Upon receipt of any prize, each winner is required to comply with any and all applicable federal, state and local tax that becomes due with respect to the prize(s). 8. General Rules: By participating in this Contest, Entrants acknowledge compliance with these official rules including all eligibility requirements and agree to be bound by these rules and by the decisions of Sponsors, which are final in all matters relating to this Contest. RN is not responsible for any problems or technical malfunctions of any telephone network or lines, computer online systems, servers or providers, computer equipment, software, failure of email or software due to technical problems or traffic congestion on the Internet or at any Web site or combination thereof, including injury or damage to participants or to any other person's computer related to or resulting from participating in or downloading or uploading materials in this Contest. If, for any reason, the Contest is not capable of running as planned, including infection by computer virus, bugs, tampering, unauthorized intervention, fraud, technical failures, or any other causes beyond the control of RN which corrupt or affect the administration, security, fairness, integrity or proper conduct of this Contest or for any reason RN deems it necessary, RN reserves the right at its sole discretion to cancel, terminate, modify or suspend the Contest. Should the Contest be terminated prior to the stated expiration date, notice will be posted on realonearcade.com. Games that do not meet the Entry Requirements, have been released to the general public for distribution in any form prior to August 30, 2003, or have been tampered with or altered are void and shall be immediately disqualified. ANY VIOLATION OF THESE OFFICIAL RULES BY ANY ENTRANT MAY RESULT IN SUCH ENTRANT'S DISQUALIFICATION. By entering the Contest, participants agree to be bound by these Official Rules. All decisions by RN shall be final in all matters relating to this Contest. RN shall not be in any way liable for damage, loss or injury resulting from computer malfunctions, misdirected or incomplete entries or acceptance and use of the prizes. Entrants assume all liability for its participation in this Contest. All legal responsibility for proof of legal rights regarding Games lie solely with Entrant and Entrant hereby indemnifies and holds RN and all of its employees, partners and assignees harmless against any claims made in dispute of such rights. 9.Winners. For a copy of the winners list, send a SASE to: RealNetworks, Inc. "Game Developer Showdown," 2601 Elliott Avenue, Seattle, Washington 98121, attn: RealOne Arcade, or send an email request to devchallenge@real.com. All requests must be received by September 30, 2003

RealOne ARCADE

INDUSTRY WATCH

KEEPING AN EYE ON THE GAME BIZ | *everard strong*

Nintendo cuts royalty rates. Nintendo Co. Ltd. has lowered its royalty rates that third-party publishers must pay the company. The company hopes this move will help lure more titles for the Gamecube at a time when Nintendo lowered its current fiscal-year outlook. The company expects net profit for its current fiscal year to be approximately $548 million, lower than the initial estimates of $665 million. The company attributed low Gamecube sales for the lower profits.

Mythic infusion. Mythic Entertainment, developers of the MMORPG DARK AGE OF CAMELOT, has received a $32 million investment from TA Associates, a private equity and buyout firm. The company plans on using the cash to expand their online player base and create new titles.

Interplay's full-year results down. According to year-end reports, Interplay's 2002 revenue was $44 million, a 22 percent drop from 2001's revenues of $56.4 million. However, the company ended the year with a net income of $15.1 million, compared to the previous year's $46.3 million net loss. The company attributed a $28 million gain to its sale of Shiny Entertainment to Infogrames.

Mythic Entertainment, developers of DARK AGE OF CAMELOT, received a $32 million infusion.

Acclaim saved from within? Acclaim CEO Greg Fischbach and senior executive vice president James Scoroposki fronted $1 million each of their own money in hopes of keeping the publisher from being delisted from the NASDAQ due to low share price. In exchange for the cash infusion, the two executives were each issued 2 million shares of company stock, plus other warrants.

ATI revenues up, but profits down. Blaming a decline in royalty income, ATI Technologies' second-quarter earnings dipped slightly to $318.5 million from last year's $322 million. Charges in the quarter included an $8 million class-action lawsuit settlement, $2.8 million in the closing of a European manufacturing operation, and other expenditures.

Xbox cuts European prices. In a move calculated to position itself ahead of Nintendo, Microsoft lowered the price of its Xbox console in the European market. The move, Microsoft's third European price cut in less than a year, makes the console cheaper than the PS2 and about the same price as the Gamecube.

Send news items and product releases to news@gdmag.com.

THE TOOLBOX

DEVELOPMENT SOFTWARE, HARDWARE, AND OTHER STUFF

Two 3DS Max plug-ins from Turbo Squid. Turbo Squid has released AfterBurn 3 and Kaydara's HumanIK plug-ins for Discreet's 3DS Max. AfterBurn 3 enables rendering of realistic effects ranging from clouds, dust, and explosions to liquid metals. HumanIK's features include automatic character rigging, a BodyGenerator Max script, and other features. www.turbosquid.com

Maya 5 unveiled. Alias|Wavefront announced the newest version of their 3D modeling and animation program, Maya 5. The new version features four rendering options — the Maya software renderer, Mental Ray for Maya, a new vector renderer, and a hardware renderer — plus enhanced character animation tools, new modeling tools, API updates, new data exchange options, and expanded polygon mesh-editing tools. www.aliaswavefront.com

New game editor development tool announced. Adventurerland Entertainment has announced its Lab Technology Construction Kit, software that includes a built-in compiler/linker that creates and deploys a game engine's output. The kit supports DirectX, OpenGL, and SDL. www.adventurerland.com

Softimage reveals Softimage|XSI 3.5. Softimage announced the newest version of their modeling and animation environment. Version 3.5 features automatic symmetrizing of polygons, flattening of UVs, more realistic hair generation, and UI support for Cg and DirectX vertex and pixel shaders. www.softimage.com

UPCOMING EVENTS CALENDAR

CHRISTIAN GAME DEVELOPERS CONFERENCE
CASCADE COLLEGE
Portland, Ore.
July 25–27, 2003
Cost: TBA
http://cgdc.graceworksinteractive.com

SIGGRAPH 2003
SAN DIEGO CONVENTION CENTER
San Diego, Calif.
July 27–31, 2003
Cost: $50–$950
www.siggraph.org/s3003

CLASSIC GAMING EXPO
JACKIE GAUGHAN'S PLAZA HOTEL
Las Vegas, Nev.
August 9–10, 2003
Cost: $35 weekend pass
www.cgexpo.com

bigWORLD™

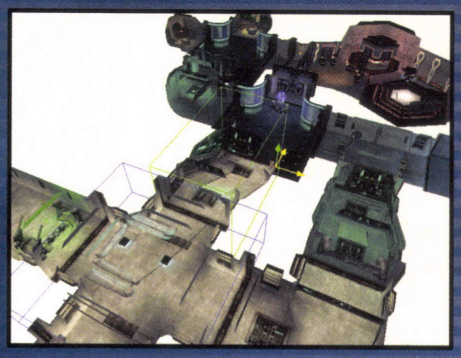
The Complete MMOG Solution™

E3 expo

South Hall
booth #436

The BigWorld™ Technology is the complete solution for building the next generation of Massively Multiplayer Online Games

Dynamic load balancing in action



Advanced world editing tools

Client Engines - PC available now. XBox in development. Includes DirectX 3D engine, scripting engine and GUI.

Server Backend - distributes game events between all clients and manages world data.

Tools - a set of tools to build and manage your world.

BigWorld™ is the only engine that can deliver all of this:

- A uniquely scalable, customisable and fault-tolerant server infrastructure that can handle millions of players
- Extensive LOD controls on characters, scenery objects, and transmitted data to allow massive crowds
- Polished tools for building worlds fast
- Fault tolerant and dynamically reconfigurable servers
- Multi-Platform PC, XBox, PlayStation2 and more
- Powerful server management and monitoring tools
- Responsive support with engineers dedicated to doing everything possible to ensure the success of your title

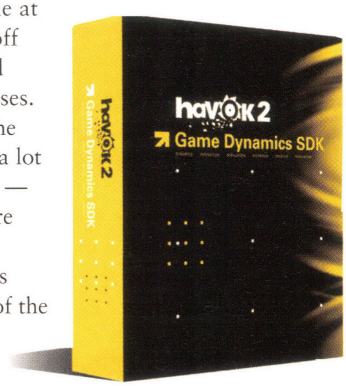
Phone : +61-2-6162 5120
Email: licensing@bigworldtech.com
www.bigworldtech.com

Havok 2: All Rag-Dolled Up

by justin lloyd

"Does it do anything but driving games?" is a question that draws groans and smiles simultaneously from the Havok team, makers of their eponymous game physics middleware that recently received a major upgrade to version 2.0.

In response to what must be an all-too-frequently asked question, Havok 2.0 broadens its scope to include character physics much more prominently over earlier versions. In that sense Havok 2.0 is not an upgrade, it's a whole new product. New demos, viewable at www.havok.com, show off this functionality to good effect over previous releases.



New and improved. In the original Havok we heard a lot about dynamic characters — rag doll — but it was more akin to having a dynamic corpse. Havok 2.0 extends this concept, losing a lot of the dead-body feel of its predecessor. I'm not an animator by any stretch but I was able to develop my own walk cycles and attach the Havok physics system and character control far easier than with earlier versions. Havok has realized the need to dictate full control over character look and movement in the game, both for player and nonplayer, because that's what the game player wants. I don't always require accurate physical simulation, for example when a more cinematic or fun-focused effect is desired. Havok attempts to provide this control, and while far from perfect, it's a step in the right direction. I was able to emulate parts of the standard zombie demo easily enough that as my animation walks forward I "detached" the head, allowing it to loll freely. When hitting the character with an impulse, such as a simulated shotgun blast, I switched state to playing a "knocked back" animation, detached the arms, and let the physics control the flailing arm movement and torso rotation. This could have been extended to letting arms dangle limply and legs drag heavily along the ground when targeted. I was also happy to see a proper section in the Havok manual for handling characters and control, so you aren't left floundering.

You may not have heard this before, but Havok does driving games! There is new functionality in the vehicular component of the SDK, but for me, it was important to see how easy it was to create vehicles now. The original Havok Vehicle SDK provided a lot of black-box functionality that was poorly documented and nonintuitive. With earlier releases, I spent a lot of time spinning my wheels, literally, trying to figure out how to balance all the forces. Havok now assumes that you don't inherently know what anti-roll bars are for, how to tune the suspension based on terrain and vehicle type, or how to calculate friction circles for cornering. Havok details specifically how their Vehicle SDK black box works and assumptions are made, removing a lot of past guesswork from the equation.

Screenshot from the author's second demo, demonstrating blending animation and physics.

Another thing that changed was automatic construction of object groups. Havok 2.0 introduces "Islands" so that you no longer have to take an educated guess as to in which group to place your object so that the collision system is optimal. Islands are automatically constructed at run time, containing small groups of objects based on the simulation criteria. I had a previously existing prototype inside of which was a spiral of dominos that could be knocked over. Originally, I placed all of the dominos in their own group, which was fine, so long as the player didn't attempt to knock over multiple dominos at once, causing multiple cascades, as that would slow down the simulation. Havok 2.0 took care of the grouping for me; once a few dominos

JUSTIN LLOYD | *Justin has over 18 years of commercial game programming experience on almost every released platform.*

YOUR MILEAGE MAY VARY: HAVOK 2 USERS WEIGH IN

"We've had a positive experience with Havok, although because we're about halfway through our current game life cycle, we still have issues that we need to resolve. If we are happy with Havok at the end of the game, I can certainly see us using it again, as much of the hard work of learning the API will already be done. The documentation could be better though; a real boon would be to have more code examples in the docs themselves, rather than having to track down the usage inside a demo."

— Mark Baker, Mucky Foot

"Even with Havok it still takes significant work to combine gameplay and physics in a way that they enhance each other rather than detract from one another. When we originally looked at physics middleware a year or so ago, Havok looked like the most complete product on all platforms."

— Sam Baker, Paradox

"We aren't using the newest release yet, but during development we've given them lots of feedback and many of those issues have been addressed in Havok 2."

— Jay Stelly, Valve

came to rest, they formed their own "simulation island" and became quickly deactivated. This relieves a huge burden from me having to take what was, at best, an educated guess as to which group objects should be placed in. I also no longer need to verify that someone else has placed objects in the wrong group.

Other platforms. I've only been able to work with Havok 2.0 on VC++ 6.0 running under Windows XP. Circumstance did not permit me to compile on Sony and Nintendo dev kits, so my only experience with Havok 2.0 on those platforms is playing with the interactive demos at GDC. From what I was able to glean from their new Visual Debugger, a module similar to the real-time VTune, the frame rates on complex scenes — such as the zombie demo with a dozen animating rag dolls — and character control were rock steady. CPU and memory usage never spiked above 50 percent, usually hovering around 20 percent, and when it did climb it was only for one or two frames when multiple characters attempted to interpenetrate. This was a pleasant surprise, as in the past with Havok 1.7 I could bring my 3GHz, 512MB, 64MB Radeon 9000 to its knees with a half-dozen rag dolls. Looking through the documentation and API headers reveals that Havok has gone to great lengths to ensure that each platform is properly targeted, the PS2 build making good use of the scratchpad and vector units.

How useful Havok's Visual Debugger will prove is questionable. It certainly gives you insight into the work being performed by the physics engine, but I'd like to see more hooks to allow integration into standard profiling tools such as VTune and ProDG's tools for Gamecube and PS2. I've worked on projects where the team has provided a profiling solution, and my experience with the Havok VD suggests that it does not provide the hooks that will be needed.

Support. Unlike a few middleware companies I have dealt with, Havok goes to great lengths to ensure that the people providing tech support for the licensees are qualified to do so, ex-game and physics programmers to a man. Their tech support is second to none. From prior real-world experience with the 1.8 SDK, technical queries submitted late in the afternoon would elicit a response by the time I returned to my desk the next day.

Havok seems to understand that as a middleware provider, customers' project deadlines don't wait for them. Pertinent questions that would apply to other developers are often placed in the FAQ — even if they are only asked once — along with useful code snippets; even small features are quickly rolled back into the main Havok code base. I saw just how quickly this happens when attempting to create a vehicle that behaves as the Warthog in HALO, with all four wheels steerable. Within 18 hours I had my answer, and a day later it was a FAQ and code snippet on how to modify

THE TESTBED

Each project is different, and a physics simulation, like Internet traffic, changes moment to moment, making it difficult to provide hard data that you could form a judgment around. My test rig was a 3GHz, 512MB DDR 2100, 64MB Radeon 9000–equipped notebook computer running Windows XP and DirectX 9.0. I assembled two demos for this review.

The first test project consisted of approximately 150 objects in close proximity, 10 of which were considered complex; small-scale versions of the St. Louis Arch that were pinned, i.e. infinite mass and immovable. The rest of the objects were simple geometric shapes (boxes, wedges, and spheres) that the player could interact with in varied fashion. There were also a half dozen rag dolls consisting of 15 bones each; a rag doll was connected to its immediate neighbor via 2D dashpots, forming a loosely coupled chain. The player was represented as a single, two-wheeled vehicle that rode like an old-fashioned bicycle, constructed with one large wheel at the front and a single small wheel positioned just slightly behind for stability. Due to the small distance between the wheels, the vehicle possessed a very small turning radius, enabling the player to negotiate the tight environment.

The second demo — a simultaneous blending of animation and physics — consisted of two dozen rag dolls animating a walk cycle that the player could shoot at with the mouse. The quoted memory footprint covers the entire demo, including graphics.

	Peak Frame Rate	Average Frame Rate	Peak Memory Usage	Average Memory Usage
Demo 1: Large, 150-Object Scene	120	80	10.1MB	9.4MB
Demo 2: Animation and Physics	174	98	2.9MB	2.7MB

★★★★★ excellent
★★★★ very good
★★★ average
★★ disappointing
★ don't bother

HAVOK 2.0 ★★★★½

STATS

HAVOK
www.havok.com
San Francisco, Calif.
(415) 543-4620
Dublin, Ireland
+353 (1) 677-8705

PROS

1. Flexible licensing.
2. Very responsive, technically savvy support and sales people.
3. The development team is Irish.

CONS

1. Character control isn't perfect.
2. Poor documentation for such an in-depth product.
3. Visual Debugger needs better integration with other tools.

the Vehicle SDK. Once you move beyond evaluation, Havok supplies 95 percent of the source code, saving valuable time when you are hunting down an elusive bug or attempting to optimize your game. Havok is quite happy to let you look inside their physics black box as much as you like.

Documentation. I'm of two minds concerning the way Havok pursues documentation. The learning curve for the Havok SDK can be steep, one reason being that Havok relies more on demos than documentation to illustrate a point. That causes two learning bottlenecks: First, you have to spelunk through many, many source files looking for the demo or snippet of code that you think you need (the "I'll know it when I see it" approach). Second, because the Havok demos are built around a common framework of code, the body of which pulls in every feature that Havok provides, doing this requires more support functionality. You have to build up a mental roadmap of that in your head, learning what's important and what's immaterial to the particular demo before you can delve in to the small piece of code that you really should be concentrating on. The demos are good starting points, but better documentation showing the bare minimum needed — and the reasons why these steps are needed — would make the job easier.

Terms. Licensing terms vary based on numerous factors, such as the number of titles and platforms involved, and there are various support options from which to choose. One constant is that license fees are one-time-only, with no royalties involved.

Final word. The star rating I assigned for Havok 2 represents my best stab at mapping my experience with a very complex product to a simple, five-point scale, which isn't a neat fit, especially given how different real-world game production needs and environments can be.

The bottom line is that Havok is maturing. Version 2 takes great strides forward beyond Havok's driving-game roots, especially in the realm of character-based physics, making it worthy of evaluation by developers on a wide range of projects.

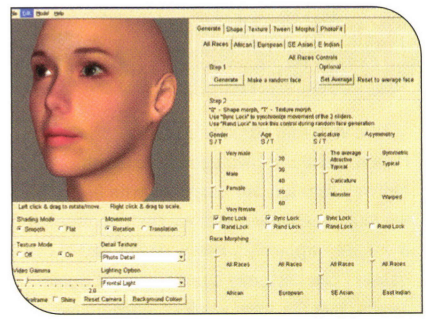
Singular Inversions' FaceGen Modeller 2.2

by michael dean

FaceGen Modeller 2.2 is the newest version of Singular Inversions' face and head creation software. It has been designed to allow a user to create custom, unique faces faster than traditional 3D modeling packages typically allow.

Getting started is easy and fun. Upon launch, a default head loads into the shaded viewport, and I was immediately able to create random faces with just a push of a button. There is also the option to load faces that have been previously custom-created in the software.

To customize the premade and randomly generated heads, FaceGen comes complete with a simple yet powerful modeling toolkit. It differs from a traditional 3D modeling package in that geometry is not directly manipulated on the vertex/face level but through a series of sliders that control all aspects of your model. For example, if I create the face of a young woman but want to have the model reflect an older age, rather than push and pull vertices, I use two sliders that control age, one for the geometry and one for the texture. Move them forward, and cheeks lose their fullness, the nose grows, and the skin weathers, all based upon the face's natural aging process. It works remarkably well, which is how everything in FaceGen functions. There are also sliders to control masculinity and femininity, race, symmetry, and realism.



A custom head created in FaceGen Modeller from a few simple snapshots of a subject. No additional editing was needed, and the default result was amazingly accurate.

Taking this to a deeper level, there are sliders that make up, and append, subsets of these more general categories. The user can go into a much more detailed slider group and fine-tune features like the character of a nose. Users can also adjust for a heavier brow or make a longer face, for example. The ability to easily create faces that differ greatly from one another doesn't disappoint. Morph targets can be generated using combinations of emotion sliders, as well as phoneme sliders.

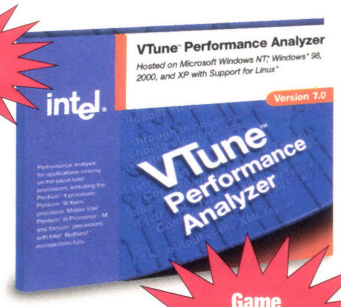
FaceGen's built-in faces are customizable, and there's plenty to do with the templates and modification tools provided. However, many artists will want the ability to create heads from unique material, which is where the PhotoFit service comes into play. With PhotoFit, simple snapshots of people can be made into full models, all fully modifiable, as they are with the template faces. After the images are acquired, the wizard interface brings them into the software and the images are sent via Internet to Singular Inversions,

Two Clicks to Better Performance!

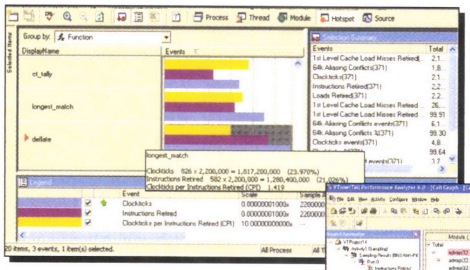
Intel® VTune™ Performance Analyzer 7.0

Intel's VTune™ Performance Analyzer 7.0 helps you increase your software performance by enabling you to locate and remove bottlenecks in your code. In addition to its powerful analysis features and easy to use GUI, version 7.0 now offers seemless integration into Microsoft* Visual Studio .NET*!

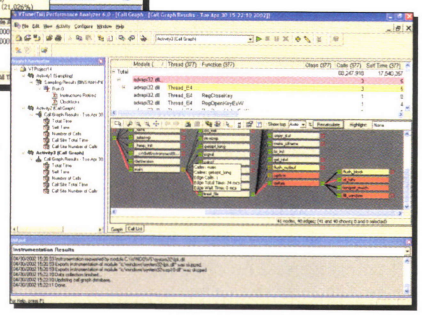
New Version 7.0



Game Developer 2002 Frontline Award Finalist



Time & Event Sampling
View performance by time or processor events (cache misses, branch mispredictions, etc.).

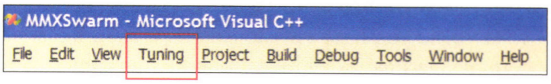


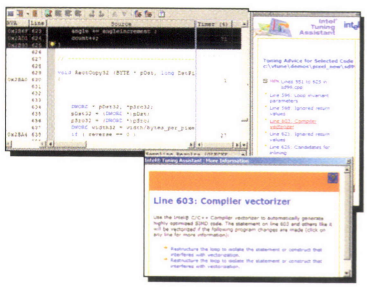
Call Graph Profiling
View call paths, the number of times a function is called, time spent in each function and more.

View the Demo

Intel's demo will show how in just two clicks you can be directed to the lines of source code most affecting your performance.

Integrates cleanly into Microsoft* Visual Studio .NET* development environment.





Intel Tuning Assistant
Uses Intel compiler technology to give advice on improving code performance.

> ***"The Intel VTune Performance Analyzer took a multi-day task and turned it into a sub-day task."***
>
> **—Randy Camp**
> *V.P. Software Research and Development*
> *MUSICMATCH, Inc.*

VTune™ Performance Analyzer 1.0 for Linux

The award-winning VTune™ Performance Analyzer is now available for Linux! Find the hotspots and bottlenecks in your Linux* applications, drivers and kernels.

Intel® VTune™ Enterprise Analyzer

VTune™ Enterprise Analyzer helps you locate and remove performance bottlenecks in distributed web applications. Supports multi-tier Java* (J2EE) and .NET applications.

YOU SAVE UP TO $1,485!	Paradise #	Retail	Discount
Intel® VTune™ Performance Analyzer v7.0	I23 0A2N	$699.00	$534.99
Intel® VTune™ Performance Analyzer v1.0 for Linux*	I23 01A0	$699.00	$534.99
Intel® VTune™ Enterprise Analyzer v2.0 .NET Edition	I23 0A51	$12,500.00	$11,014.99
Intel® VTune™ Enterprise Analyzer v1.0 Java Edition	I23 0A54	$12,500.00	$11,014.99

Download the Demo and a Free Trial at:
programmersparadise.com/intel/gdm

Programmer's Paradise®
To order, or request additional information call:
800-423-9990
Email: intel@programmers.com

© 2003 Intel Corporation Intel, the Intel logo, Pentium, Itanium, Intel Xeon and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. *Other names and brands may be claimed as the property of others.

03CT06C

★★★★★ excellent
★★★★ very good
★★★ average
★★ disappointing
★ don't bother

two photos are supported) into a FaceGen file, which can be loaded into Modeller, edited, applied to a mesh, and exported as a model with UVs and texture. The results are amazing, looking as if they came from a 3D scanner. This additional feature is fast; it took about 20 minutes to receive custom-generated FaceGen files. PhotoFit also requires an additional investment, starting at $9.50 per face, after an initial 10-face credit.

FaceGen exports to the native formats of most popular 3D packages. The models are a bit heavy on geometry, but upon import into your favorite 3D package, detail is easily reduced. I converted a 6,000-polygon FaceGen head into a very nice 550-polygon 3DS Max head by using only Max's built-in MultiRes and then flipping a few edges. The entire process of creating this head took 10 minutes, with excellent results.

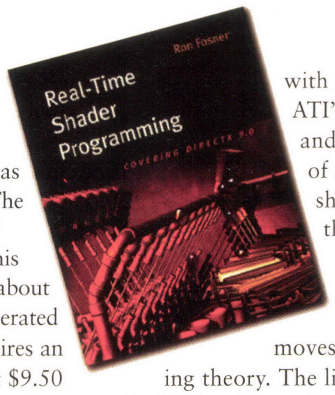
Even if the software cannot create exactly the head you have envisioned (for example, getting the bulbous nose I envisioned for a character proved impossible just using FaceGen), rest assured that FaceGen will give you a very solid starting point. At $495, FaceGen is a robust package with a lot going for it, and an artist could easily save hundreds of hours in creating of unique faces for any given project.

★★★★ | FaceGen Modeller 2.2 | Singular Inversions | www.singularinversions.com

Michael Dean is currently an artist at Ion Storm in Austin, Tex.

Real-Time Shader Programming by Ron Fosner

reviewed by jeremy jessup



In *Real-Time Shader Programming*, Ron Fosner describes the essential elements necessary for developing shaders in a very approachable full-color book that spans just over 400 pages. Your $49 also gets you a CD with a beta version of ATI's RenderMonkey and coded examples of many of the shaders discussed in the text.

Beginning with elementary vector math, the book moves quickly into lighting theory. The lighting chapter highlights the mathematical approximation of physically based lighting using the traditional ambient, specular, diffuse, and emissive colors in a scene. Representations for reflection and refraction are derived from Snell's law and Fresnel equations. Finally, non-photorealistic rendering (NPR) from cel shading, tonal art maps, and hatching is covered through pictures and a wealth of external references. The chapter makes for an enjoyable read by providing an understandable background to lighting techniques for non–seasoned graphics programmers.

Fosner describes how to set up the DirectX pipeline to use shaders. While he touches on some of the nuances you're likely to encounter, the DirectX section seemed a bit sparse compared to the earlier chapters. The DirectX setup calls specific to shaders were well documented; however, the chapter didn't dwell on creating the pipeline.

The book then describes several current shader creation and visualization tools. This chapter is relatively short, perhaps due in part to the newness and hence volatility of cutting-edge shader tools. While high-level shader tools, such as Nvidia's Cg and Microsoft's High-Level Shading Language, were briefly mentioned, the focus was on the shader language primitives. As such, it provided a sound fundamental shader approach universal to all higher-level shader implementations.

With the groundwork firmly in place, a wealth of shader examples follows. Starting with the minimal vertex shader, additional functionality is layered to build more complex shaders. Sample shaders are developed using the lighting equations presented earlier. While it may take a little time to digest some of the more sophisticated examples, such as the cartoon shader, the text provides adequate descriptive detail coupled with helpful color pictures to make it easier.

The final chapter provides a vertex and pixel command reference. Each command describes the supported shader version, usage, and a short example. The book covers shader implementations for both DirectX 8.x and DirectX 9. Differences between the two versions are noted throughout the sample code and reference section. When appropriate, additional notes on specific DirectX versions are also provided, and Fosner does a good job of providing references throughout the book for further information on a subject.

While having familiarity with the rendering pipeline, I found this book very approachable and easy to understand despite not being a low-level graphics programmer. The writing and companion tools provided challenged me to explore the world of shaders and attempt to write some of my own. The tools were a great aid, since they freed me from having to write my own engine and instead let me focus on the actual shader code. Writing in pseudo-assembly may not seem like fun, but it was — especially when I could experiment with one of the precoded routines Fosner supplied and view the results of a vertex or pixel shader routine through RenderMonkey instantly.

Shaders will play an increasingly important role in game development, and Fosner's book presents introductory groundwork necessary for developing custom shaders. For the experienced shader programmer the book's depth may not satisfy, but to those new to shaders or want to experiment with different rendering effects, this book is a great place to start.

★★★½ | *Real-Time Shader Programming* by Ron Fosner | Morgan Kaufmann | www.mkp.com

Jeremy Jessup is a programmer for Rockstar San Diego.

Next time they tell you that hitmakers aren't interested in games, maybe they should talk to him.

And while they're at it, they should also talk to Crystal Method, Avril Lavigne, Snoop Dogg, Marilyn Manson, Megadeth, Sum 41, Outkast, DMX, Judas Priest and all the other chart-topping artists and writers from EMI Music Publishing who are licensing hits to the world's hottest games.

emi musicResources

When it comes to music, we're the name of the game.

1-866-EMIGAMES • games@emimusicpub.com

No One Markets Beer Forever

Monolith's Samantha Ryan

Before joining the game industry in 1996, Samantha Ryan had a solid 10-year career in broadcast marketing, working on projects for Infinity Broadcasting, Miller Brewing Company, and Frito-Lay. Ryan joined Monolith in 1998 and was recently promoted to president. Samantha's roles are varied, from pursuing new projects and partnerships to overseeing trademarks and legal agreements. "Conversely," she admits, "if the conference room is messy, I'll clean it up."

We caught up with Ryan to find out how she leads a company like Monolith, and how her marketing background comes into play when dealing with videogames.

After 10 years in broadcasting, Samantha Ryan decided it was time for a real career and joined the game industry. She is now Monolith's president.

Game Developer: How has your past experience in entertainment marketing crossed over into your work at Monolith? How does it affect the way you lead the company?

Samantha Ryan: It's true my past has been a little unusual, although that doesn't seem to be uncommon in this industry. For 10 years prior to entering the game industry I worked in broadcasting in both a production and a marketing capacity. The knowledge I picked up there has definitely shaped my approach to developing games. For example, there's something about the marketing process itself that is incredibly intriguing. I'm not talking about creating box art or ads, although those are certainly challenging. Rather, positioning and the psychological aspects of team dynamics as well as consumer marketing are fascinating subjects and are worth studying by anyone in senior management, regardless of duties or title.

GD: When developing a game idea — like No One Lives Forever — how important is the brand's marketability, in addition to its gameplay?

SR: A brand's marketability is an aspect of development that was de-emphasized in our early years. However, by this stage in our maturity as a developer, it's become very important. We work closely with our publishers on each title's overall positioning as well as the ongoing marketing and publicity efforts.

When developing a new intellectual property, we strive to create a robust universe populated with compelling characters. The more well-rounded your property, the better it will lend itself to application in other mediums. No One Lives Forever taught us a great deal, both things we did right and things we need to do better.

GD: With three teams working on different projects under one roof, how do you balance out projects, personnel, and other resources so that not everyone is in crunch mode all the time?

SR: Over the years we've learned to carefully select and schedule the timing of our projects; typically we prefer no project finish within six months of another project. At both the beginning and tail end of large projects, where you don't need a full production team, people tend to move around to help on other titles. This strategy allows us to keep our teams and personnel together for the bulk of a project, but also to give people some variety, and test them out on other teams.

GD: What has Monolith done to create an environment where its employees can have a sense of fun with their work and stay motivated while still being able to meet milestones?

SR: This is a difficult challenge for any management team. A great environment starts with great projects, so we're pretty selective. In addition, we plan company events, such as movie screenings, outdoor parties, and the like. We have also chosen an office-based seating plan rather than a cubicle-based arrangement; we find people enjoy having a bit more privacy, and that communication isn't hampered by it.

GD: How do you keep your creative teams creative when they are working on licensed content and sequel material?

SR: Staying creative on these types of projects has never been a problem for Monolith. It's all about the attitude with which you approach these efforts. In addition, we've had some wonderful properties with which to work: Aliens vs. Predator, Tron, and The Matrix.

GD: What is the key to retaining your best employees?

SR: The best projects. Certainly maintaining competitive benefits and salaries are important, but most Monolith employees, including myself, work in this industry because we truly love gaming. Therefore, finding great projects for our teams is probably one of our most important priorities. This is no easy task, but definitely worth the effort.

GD: You had most of the team from NOLF return for NOLF 2. How do you encourage a team to keep a fresh approach while building on a franchise?

SR: Because it's usually right at the end of a project that a game finally takes shape, we believe a sequel is an opportunity to refine and polish all the concepts developed for the first game. Also, as the team grows from one project to the next, new hires or transfers from within Monolith are expected, and each new person brings a unique perspective to the existing team.

Shanan (and the rest of our team) would be happy to help you configure your Anthro Workstation.

If you can imagine it, we can configure it.

Whatever your heart (or office space) desires, Anthro can come up with a furniture solution just for you. Everything from Anthro is flexible, modular and strong. Strong enough to come with a Lifetime Warranty. And with over 75 different accessories to choose from, and hundreds of possibilities, you can always reconfigure your Anthro furniture to fit your changing needs. Imagine that.

For a free catalog, contact us at ANTHRO.COM or 1-800-325-3841.

ANTHRO®
TECHNOLOGY FURNITURE®

© Copyright 2003 Anthro Corporation

Nokia N-Gage™ Mobile Game Deck Connects With Developers

Game developers had the opportunity to meet with Nokia executives at GDC 2003 to discuss the Nokia N-Gage™ mobile game deck. Forum Nokia interviewed Rob Milne, director and general manager of the Nokia N-Gage device publishing group, and Dr. Jonathan Sharp, senior manager, games publishing, Nokia, to find out about the technical and business opportunities that are available to developers.

FN: Why is Nokia getting into the publishing business in the first place?

Sharp: By publishing our own titles to the N-Gage platform, we gain a deeper understanding of game design. Also, because we have the first scope on new technologies, we can experiment with them earlier, and look at their usefulness for gameplay.

FN: Is that partly to help jump-start the business?

Milne: As an example, one strength of this device is its wireless connectivity – Bluetooth, the cellular connection, and so on. We need to have titles that exploit these capabilities. Nokia-published titles will be leading that.

FN: Does what you learn feed back to the teams that are developing the tools?

Milne: Absolutely. That's one of the other big benefits. The SDK is constantly evolving. It will have new APIs and some new functionality integrated into it constantly.

Sharp: Also, there are some unique Nokia sponsorships that are suitable for games, such as the relationship between Nokia and FIS World Cup Snowboarding and the Nokia Sugar Bowl.

FN: What sort of demographic are you aiming for?

Milne: We're targeting young adults, as opposed to the children's market that other handheld devices cater to.

FN: If I'm a games developer interested in producing titles for Nokia, what should I do?

Sharp: Toward the end of last year, we issued a request to Forum Nokia developers to submit concepts for the Nokia N-Gage device. We gave (developers) very few guidelines, but we asked them to think creatively. We had more than 80 submissions, which was an amazing number. The FN game experts winnowed that down. At the beginning of January, at the concept submission meeting, which we run every month, the FN team presented nine of the top concepts from that initial submission. From that, we selected one title to go into development. We can't discuss the title or who the developer is, but it is an original game and something that we're very excited about.

FN: Once you've been vetted and invited to work on a game for the Nokia N-Gage device, what's the process?

Sharp: Once we've approved a concept, we go back and have a discussion with the company to understand the actual business proposition and try to understand the game concept more fully. Then we perform an assessment of the company to understand its capabilities and how we can work together.

The Nokia N-Gage Mobile Game Deck

Key Features:
- *104-MHz ARM processor*
- *Shared Memory up to 4 MB + MMC*
- *Flash memory up to 2.8 MB without MMC*
- *Built-in camera*
- *Based on Symbian OS v6.1*

UI Features:
- *176 x 208 pixels*
- *Color depth: 4096 colors, 12 bit*
- *WAV-format support via the Media-Server API*
- *MIDI engine available*
- *Mono and stereo output*

Java™ Features:
- *CLDC 1.0*
- *MIDP 1.0*
- *NOKIA UI API*
- *JSR 135*
- *JSR120*

Peer Connectivity:
- *Bluetooth*
- *USB*

Browsing Features:
- *XHTML*

Messaging Features:
- *MMS*
- *SMS*

© Nokia 2003

FN: Does that also include looking at its financials to make sure it can carry the project to completion?

Sharp: Yes. Then, we're into our game development process, determining deliverables, actions, inputs and outputs, and milestones, all the way until the product ships. Provided that the process is acceptable and we can find a way of working together, we then move toward a legal agreement and then into actual development.

FN: Do you look for firms that already have experience on small form-factor screens?

Sharp: That's a positive, but we don't limit ourselves to just those firms. The real thing that we're looking for is games experience. That's the most important thing.

FN: What about multiplayer experience?

Sharp: No, just general innovation. It doesn't necessarily have to be multiplayer, although we do want to use the connected capabilities of the device.

FN: So once a developer is "inside" and has begun the process, what kind of support do you provide?

Sharp: There is a specific Nokia N-Gage device software development kit, with a whole series of optimized tools for game development and documentation to help game developers. We supplement that with first-line developer support.

FN: By "first-line developer support," do you mean someone who answers the phone and talks to you in person?

Sharp: We do that, yes.

FN: How is the Nokia N-Gage device SDK different from the Series 60 SDK?

Sharp: It's based on the Series 60 SDK, but we've added a number of APIs to help with some of the features that we think are very important in games. We're looking for high-performance rich games, and to support that, we've provided some additional functionality.

FN: Do you offer any kind of development funding to developers after you've vetted them?

Milne: We are following on a case-by-case basis the normal business practices of the games industry, meaning providing development funding as a recoupable advance against future royalties. Not for every title, however.

For information about developing for the Nokia N-Gage mobile game deck and other Nokia mobile devices, as well as the Series 60 SDK and other resources, visit ***http://www.forum.nokia.com/games.***

VOLUME 1 ISSUE 3 : JUNE 2003
SPONSORED BY NOKIA

GameDeveloper Mobile

A bi-monthly supplement from the editors of *Game Developer* magazine

3D Standards Already?

By Ben Calica

Superscape's use of their 3D technology in their CHESSCAPADE game points toward a cell phone–based 3D future.

Much to the surprise, and often disbelief, of the mobile development community, there are serious efforts underway to establish 3D standards in an industry where memory is counted in the hundreds or even tens, and where floating point math seems as far off as personal rocket-cars. To many developers currently laboring to get the most basic gameplay elements into the system resources available, the idea of 3D standards makes them roll their eyes. "We're way too early," says JAMDAT president Scott Lahman.

Nonetheless, there are two major sets of official standards undertakings (not counting Microsoft's Direct3D for Windows CE). The first is an effort by the nonprofit Khronos Group to create a variant of OpenGL called OpenGL ES. The other is a Java Specification Request known as JSR-184, being led by Nokia with a wide industry support group. Interestingly, many of the same members are in both standards bodies.

Why both? "We're trying not to make the same mistakes we made on the PC," says Carl Korobkin, CTO of 3d4W, a 3D technology company, who sits on the core committees for both groups. Many in the industry have vivid memories of how competing platforms and standards blocked the PC game industry before OpenGL and Direct3D became accepted standards. 3Dlabs VP and Khronos Group secretary Neil

CONTINUED ON PAGE 2 COLUMN 2

BEN'S CALL

observations from a mobile playground

by Ben Calica

I'm Not a 9-Year-Old Boy!

Every once in a while a moment smacks you in the face and starts a whole cascade of insights. I was so smacked recently and it left me feeling like I had somehow stumbled on a clue about a couple of the groups that are most likely to buy the games we're building.

Insight Set #1: The Openwave Kid. Openwave brought out this 18-year-old to do a little color at a press event. He had been part of a test group that was given a cool phone and the chance to play with it for a few months. Though they may have intended only to have him talk for a couple of minutes, he gave us press-types the clearest window into his generation, and we didn't want to let him go.

One of the other panelists talked about how his teenage daughter bought four ringtones a month.

CONTINUED ON PAGE 3 COLUMN 3

INDUSTRY news

Wildseed Puts Some Skins in the Game

Seattle-based Wildseed recently showed their intelligent cell phone skin prototypes. Their Smart Skins include firmware to what has previously been a fashion accessory, to change the environment of the phone as well. One of their skins adds Game Boy–style controls and is aimed at turning the phone into a more game-centric device. The company is targeting release in the latter portion of 2003 with Kyocera, Curtel, and Sewon Telecom as handset partners. Wildseed was founded by DirectX co-creator G. Eric Engstrom.

www.wildseed.com

JAMDAT Opens European Office

Los Angeles–based developer and publisher JAMDAT Mobile opened a new branch office in London in April, marking the mobile game provider's first overseas foray. JAMDAT's first European product was TONY HAWK'S PRO SKATER 4, which comes preloaded on European Sony Ericsson T310 phones. Juan Montes, formerly of

CONTINUED ON PAGE 4 COLUMN 3

inside...

Hexacto: Using Lots of Lemons to Make Lemonade 2

Mobile Developer Resources 4

Hot Handsets 4

GameDeveloper
Mobile
VOLUME 1 ISSUE 3
JUNE 2003
EDITOR FOR MOBILE DEVELOPMENT
Ben Calica • bencalica@yahoo.com
ART DIRECTOR
Audrey Welch • awelch@cmp.com
A PUBLICATION OF GameDeveloper
EDITOR-IN-CHIEF
Jennifer Olsen • jolsen@cmp.com
MANAGING EDITOR
Everard Strong • estrong@cmp.com
PRODUCT AND INDUSTRY NEWS
news@gdmag.com
PUBLISHED BY THE GAMA NETWORK,
A DIVISION OF CMP MEDIA LLC
PUBLISHER
Jennifer Pahlka • jpahlka@cmp.com
DIRECTOR OF SALES
Michele Sweeney • msweeney@cmp.com
415.947.6217
CMP
GamaNetwork

feedback
editors@gdmag.com
let us know what you think!

> 3D STANDARDS CONTINUED FROM PAGE 1

Trevett echoes the sentiment: "Without an industry-standard API, the hardware vendors don't know what to accelerate. This leads to two outcomes: some hardware vendors won't bother developing embedded graphics solutions because they have no certain way of accelerating any applications, or other hardware vendors will push ahead and develop their own proprietary APIs to suit their hardware."

There is not universal support for bringing 3D standards to the wireless space, however. One poster on WirelessGamingReview.com lamented the reintroduction of the 3D technology that many have pointed to as the cause of years of stale game design. However, beyond reintroducing familiar design pathways, 3D is a potential solution to one of the greatest problems facing wireless game developers. It provides a way to deal with radically different screen resolutions and sizes by having different windows onto the same world.

The Khronos Group hopes to ratify the OpenGL ES specification in time to deliver by July's SIGGRAPH. Members include all of the current creators of wireless 3D technology and a number of the handset manufacturers, including Motorola and Nokia. But the most interesting composition of the group is a large number of the chip and core manufacturers such as 3Dlabs, ATI, Discreet, and ARM. (Nvidia was still in negotiations to join at press time.) They were working on two variations of the spec, one for full-powered devices, and the other for the lighter devices that make up the majority of cell phones. The major difference between these is the lighter spec assumes the lack of floating point math in the device. The Khronos folk go to great efforts to point out that theirs is a lower-level API and designed to be complementary to the JSR-184 spec.

3Dlabs' Trevett sums up the goal of OpenGL ES as balancing speed and hardware abstraction. "OpenGL ES, like OpenGL, is a low-level API very close to the hardware, with enough abstraction to provide portability across different hardware but low enough to provide close-to-the-hardware speed and flexibility."

A Java Specification Request, of which JSR-184 is one, is the official

casestudy

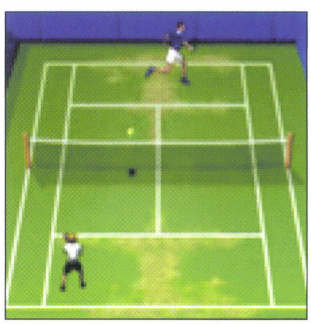
Hexacto: Using Lots of Lemons to Make Lemonade

By Ben Calica

Montreal-based game developer Hexacto is making its fortune in a lot of little bites. "When we started the company back in 2001, we wanted to position ourselves in a niche market," recalls Alexandre Tailefer, president and CEO of the company. "We wanted to create a lot of products, or at least more than one every two years," compared to traditional game development. They started with a focus on the Pocket PC, where their LEMONADE TYCOON became a hit. Despite its major success on handheld devices, LEMONADE TYCOON had an uncomfortable transition to the big screen. "The [PC game] reviewers think it sucks, but we get a lot of mail from accountants, teachers, kids, people who actually bought the PC version and who love it."

Even before their current income flow from the mobile phone side, Hexacto managed a profit off of Pocket PC and Palm. "The business has been profitable since year two, using mostly Handango.com for download over-the-air stuff [for the transactions]," says Tailefer. They've been careful to keep their cost structure small, creating a game initially and then porting it to multiple platforms. "Supporting Pocket PC alone or going after brands — paying $100,000 for a brand and spending $100,000 on development just to make one Pocket PC version — wouldn't make sense. Doing it for seven or eight platforms reduces the per-platform cost to a point where it makes sense," says Tailefer.

Hexacto's games include resource simulators like LEMONADE TYCOON (above) and sports titles like TENNIS ADDICT (below).



The company is small-device platform–agnostic, developing for Pocket PC, Palm, Symbian, Java, BREW, PC, and Smartphones. "Pocket PC was good to us, it brought us to the Smartphone platform, it brought us to the Palm platforms," observed Tailefer. "Our first big success was TENNIS ADDICT, because we used the stylus in an innovative way."

They put one to three engineers on each platform, with a dedicated resource working on the framework and specific device needs. "What we have to face is very similar to the problems with the PC. The fact that you are running on a smaller platform like the Pocket PC

3D game produced for Korean SK Telecom using 3d4W's technology.

method to get something added to Java, in this case Java 2 Mobile Edition (J2ME). JSR-184 appears to be aiming at a higher-level API than OpenGL ES. The spec lead is from Nokia, but the group also includes Sony Ericsson, Motorola, and Siemens. They are also looking to create a specification that can handle life in the small-and-light world but also scale up as the handsets become pocket rockets. There are also a number of serious chip manufacturers involved, as well as representatives from the companies that are already producing commercial 3D software for wireless devices.

The big two of those companies are U.K.-based Superscape Ltd. and the Japan-based Hi Corporation. Superscape (www.superscape.com) currently sells its services as both a wireless game developer and as a provider of 3D tools and technology to other developers as well as handset manufacturers.

Hi Corporation's Mascot Capsule division (www.mascotcapsule.com) has taken on those roles in the Pacific rim, and their core 3D technology is now on more than 10 million devices spanning most of Japan's major carriers. Although neither Superscape's nor Hi's initial efforts will be compatible with the new standards, both companies are on both standards committees and will likely have considerable say in how they evolve. (3d4W has a similar situation in Korea, with more than 2 million handsets shipping with their 3D API.)

The last interesting player in this is Imagination Technologies (www.powervr.com), who is working with three of these companies to provide 3D cores that can be embedded in phones.

The convergence elements are in place. The handset manufactures are very committed to the game space and are looking at 3D as a way to differentiate. They have the ability to include either API code sets or even 3D accelerators or cores in phones. The 3D chip manufacturers will never command the types of unit prices they get in the PC world, but mobile phone unit sales figures are at a factor that makes PCs look stagnant. And for the developers, standards present a chance to get much closer to games that deploy easily on a number of handsets. It's an area they should keep on their early detection systems.

"We're trying not to make the same mistakes we made on the PC."
— Carl Korobkin, 3d4W

only means that you need to make use of the kinds of solutions that we would have used five or six years ago," comments Hexacto engineer Dominic Marier. "We just integrated the stuff into our old framework."

With that framework in place, they don't see the need to pay for any externally developed commercial engines. "The major limitations are the fact that we don't have the hardware acceleration for the 3D rasterization and we don't have floating point," says Marier. However, he says, "This is only an issue at a low level. If you don't abuse it in the place where it is critical, you can save a lot of time. When we do get things like floating point — although I haven't heard a lot of talk about that — we'll just plug into it." Looking at what's gone before him in graphics technology, Marier says, "I don't see the point in creating new APIs. For example, when we look at the 2D rasterization APIs for the PC, we're just solving the same problems again."

Hexacto's relationship with Microsoft has borne fruit for them in a number of ways, resulting in an estimated 70 percent sell-through on Smartphones currently on the market. They have created a licensed version of Microsoft's LINKS golf game for Smartphones, and while they had to strip it down to fit in the more limited phone resources, they still had megabytes of memory instead of hundreds to work with, with a screen resolution sufficient to create a good-looking game.

While the future of the Microsoft's Smartphone offering is an open question in cell phone world, Hexacto feels that the system resources that are on the device represent a direction in which the future is headed. And when it gets there, they will have a catalog ready to go. "The industry should head to prices more like $19.95 for the games at that point," says Tailefer. "The current games can't justify more than the $3 to $6 they are charging, but as the power of the handsets increases, so should the industry's pricing." In the meantime, in the true tradition of the lemonade stand, making a little money from a lot of different customers is working just fine for Hexacto.

> **BEN'S CALL** *CONTINUED FROM PAGE 1*

Our young friend looked at him, both impressed and envious. "She's lucky, I can only usually get about two a month," he lamented. Did he want these so he could have the latest tunes? "Naw, those get way overplayed. I like classic tunes like 'Mission: Impossible' and 'Inspector Gadget'."

It seems like the big goal is to find the right tunes to match with each of his friends that will ring when they call. Over time, the most important people in his life got to get new tunes associated with them as he found better personality matches. Aha! Motivation plus social consequences. Maybe there is sense here after all.

Next question: When do you play games? I wanted to know if he played during any of the many bored moments we have in life. Sheepishly he admitted that he played mostly during class, the size of the phone making it easy to hide behind a leg or desk. I asked him if it was easier to hide than a Game Boy Advance. "Yeah. Besides, I'm not a nine-year-old boy!" The easy distaste with which he made this statement was chilling, or at least it should be to Nintendo.

Has Nintendo's longtime success in its bread-and-butter eight-to-12 market banished the Game Boy Advance to the land of little kids? There is an older generation who grew up playing videogames who are playing with their GBAs in blissful ignorance of the hipness factor, but for those in the beloved 12-to-24 market, the battle isn't between the quality of games on cell phones versus Game Boy Advance, because the GBA is, well, uncool if you are old enough to carry a cell phone.

Insight Set #2: Old fogies are running the first wave. The first sets of color phones that run Java or BREW cost in the $300 range at this point. While our previous group of 12-to-24-year-olds are a great game market, they just can't afford the new phones yet. Newer, upgraded phones are being bought by the intersection

CONTINUED ON PAGE 4 COLUMN 1

> BEN'S CALL CONTINUED FROM PAGE 3

of those who like new gadgets and those who have significant disposable income.

These qualities skew the early-adopter demographic much older. It also changes the nature of the games that will be popular in this first batch of sales: games like black jack and golf. As much as there is a general distaste for the shovelware approach, whereby all the games from before the last 10 years are being ported to phones, many of these are the games that the 30-to-43-year-old set grew up playing as kids. Do not underestimate the power of nostalgia.

For today's new generation, cell phones are hip. For the older generation, it's a socially acceptable way to bring back the fun of the past. Bit by bit this market is starting to make some sense.

Game Developer Mobile asks: "What are the hot handsets to develop for?"

Oliver Miao, CenterScore (developer): "Among the hottest handsets to develop for right now are the Nokia 7210, Motorola T720 J2ME, Nokia 3650, the Samsung A500, and the Motorola i95cl."

Justin Siegel, Jsmart (developer): "J2ME current hot handsets in the U.S. are the Nokia 6310 and 7310, and Motorola T720."

James Dierks, Senior Platform Manager, AT&T Wireless (carrier): "Nokia 3650 (Series 60 Symbian and Java), Nokia 7210 (Series 40), Motorola T720, and Siemens S56 (color). Also black and white devices of any kind — their volume in the market will reward those who take the time to deliver cool, fun pick-up games on these devices."

Mitch Lasky, CEO, JAMDAT (publisher/developer): "The top six outside Asia, in no particular order, are the Motorola T720, Nokia Series 40 (such as the 7210), Samsung Rainbow (A500/A561), Nokia Series 60 (7650/3650/N-Gage), Sony Ericsson T300, and Sanyo 5300."

Scott Orr, CEO and chief creative officer, Sorrent (developer): "Motorola T720 BREW and J2ME (number one by a big margin), Motorola i95cl J2ME, Samsung A500/N400 J2ME, LG5350 J2ME, and Audiovox/Toshiba 9500 BREW."

mobile developer resources

Downtime Reading

MOBILE ENTERTAINMENT ANALYST
Strong industry analysis focused on the wireless game industry.
www.mobenta.com

WIRELESS GAMING REVIEW
Sister publication to *Mobile Entertainment Analyst* with focus on reviews of wireless games.
www.wirelessgamingreview.com

THE FEATURE
News source for the mobile Internet.
www.thefeature.com

MICRO JAVA NETWORK
News and developer articles focused on J2ME.
www.microjava.com

***WIRELESS WEEK* MAGAZINE**
www.wirelessweek.com

MICRO JAVA NETWORK
Rich set of resources, downloads, and articles for J2ME.
www.microjava.com

TELECOMS.COM
Mobile business and technology with a world and European focus.
www.telecoms.com

INDUSTRY news

CONTINUED FROM PAGE 1

Motorola Personal Communications Sector EMEA and Sony Computer Entertainment Europe, will head up the new office.
www.jamdat.com

Metrowerks Adds Sony, Motorola Wireless Support

Metrowerks has announced support for both Sony Ericsson and Motorola handsets. The company had been adding to their wireless portfolio by not only providing the tool sets, but also acting as the developer technical support back end for a number of the handset manufacturers, including Sony and Motorola's Motocoders developer program.
www.metrowerks.com

New Handsets Galore at CTIA

Handset makers pulled the sheets off of their latest wares at the recent CTIA show in New Orleans. The main features handset makers were pushing were built-in cameras, music playback, and game support. Color screens with higher resolutions were ubiquitous, and many were supporting rockers or joysticklike input. A few were even supporting multiple simultaneous key-press for game use.

STAR DIVERSION on the NEC 515

The NEC 515 high-definition (216x162 pixels) GSM/GPRS phone ships with the impressive-looking STAR DIVERSION by Dwango Wireless. The phone is a descendent of the current generation of Japanese phones supported by DoCoMo. The phone features polymorphic sounds and multiple simultaneous key-press.

Motorola featured the E310 as their new game phone. It has a five-way navigational joystick and ties the phone's vibration feature into the sound output for simple simulated force feedback. It also ties bars of flashing LEDs on the sides of the phone to the same inputs to add more frenetic, if not less publicly subtle, fun.

Sony Ericsson showed a number of new phones, whose most remarkable feature was their large color screens. Unfortunately, the most promising of these, the P800, not only lacked multiple key-press, it lacked any game-appropriate keys at all, using a stylus as the primary input.

www.nec.com
www.motorola.com
www.sonyericsson.com

YOU CAN TAKE IT WITH YOU

GDC 2003 Audio Proceedings

Get a convenient reference to the extensive information provided at the GDC, exactly as it was presented.

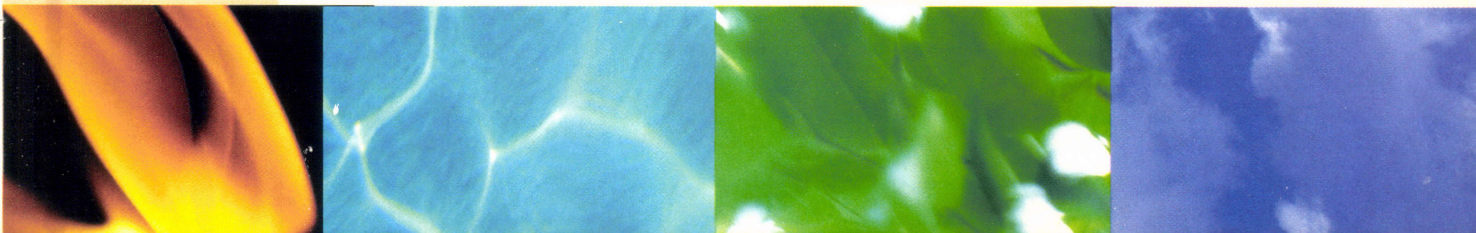
Entire Set of GDC 2003 Audio Proceedings — $399.00 - A SAVINGS OF $294

GDC 2003 Audio Proceedings individually by track: Programming, Game Design, Visual Arts, Business and Legal, Production, or Audio — $99.00

Order online at

WWW.GAMASUTRA.COM/STORE

GameDevelopers Conference



Unified Rendering LOD Part 4

In this series, we've been working on rendering large triangle soup environments. To help accomplish this we divide the environment into chunks, then create reduced-detail versions of the chunks, ensuring that no cracks are introduced in the process of detail reduction.

Last month ("Unified Rendering LOD, Part 3," May 2003), we clipped a triangle soup into two pieces, connecting the pieces with some filler triangles that I called a "seam." We created reduced-detail versions of each half of the input mesh, ensuring that the seam triangles always preserved the manifold between the halves. Now we will extend this clipping method to an arbitrary number of pieces. Then we will be able to render an LOD'd triangle soup using the chunk selection and blending system discussed in Parts 1 and 2 of this series (March and April 2003), back when the system worked only on height fields.

Multiple Clipping Planes

Last month's system only used one clipping plane. You might think that we could just apply that method repeatedly to chop up the input mesh and be done. But some complications arise, so let's look at those.

With only one clipping plane, we create only two chunks of geometry. Computing the seam between these two chunks is relatively straightforward, as we saw last month. But in Part 1 we saw that filling the gaps between each pair of chunks is not enough. In corners where multiple chunks meet, we can have a "pinhole," as seen in Figure 1. With a height field, we might fill these holes by marking the corner vertices of each chunk at preprocess time and at run time by drawing two triangles between these vertices.

But imagine trying to extend this strategy to a 3D rectangular grid of chunks. In 3D, there are two major ways that multiple chunks meet: along cube edges, where four chunks can meet, and at cube corners, where eight chunks can meet. It becomes difficult to see a way that holes can be dynamically filled because there is no longer a coherent concept of "corner vertices" to each block. A cube edge that passes through a mesh can create many "corner vertices," some of which may disappear as the chunk's resolution is adjusted. If a coherent dynamic solution exists, it's messy and probably slow. (The height field seam-drawing code from Part 2 [April 2003] already contained an unsavory amount of confusing code that performs tree traversal to find neighbors, and I'd hate to exacerbate that situation.)

So instead of filling the holes dynamically at run time, we precompute the fill patterns for these holes the same way we precompute the chunk-to-chunk seams. We expand our concept of a seam, allowing seams to touch more than two blocks. We need to precompute and store versions of each seam that cover all possible LOD states of the blocks it touches. Planning out data structures to handle the increased combinatorics for several-chunk seams is a big headache.

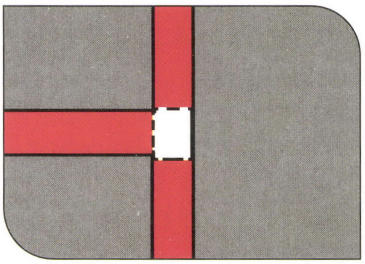


FIGURE 1. Three terrain blocks (gray with black borders) and the seam fills between them (red with dotted borders). Note the hole in the middle. These blocks are drawn with exaggerated gaps; the actual hole would be very small.

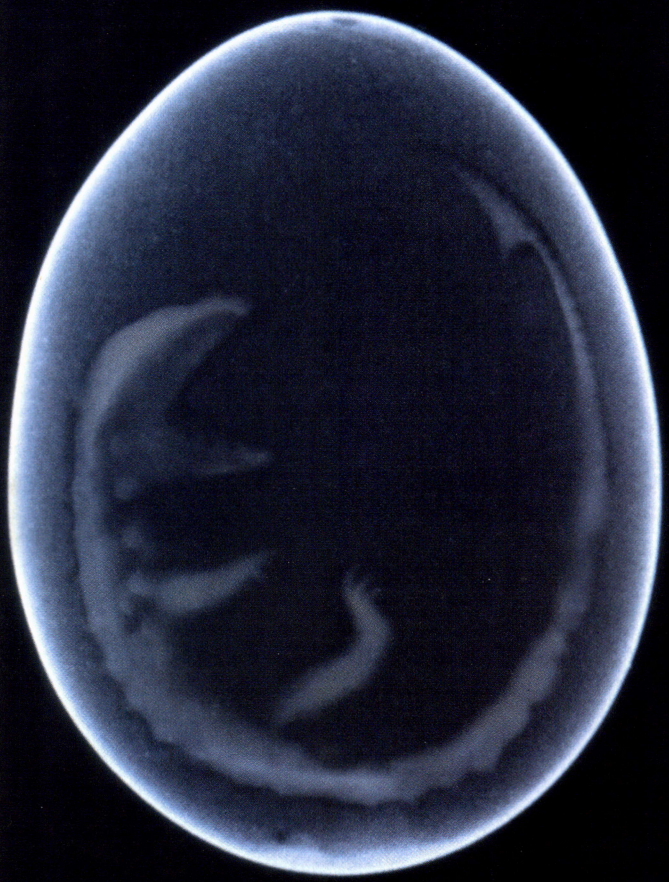
Increased Combinatorics

As it turns out, we need to contend with more than one combinatoric increase. In 3D, we usually can't impose the constraint that two fully diagonal neighbors must always be within one detail level. We'll discuss this next month, but the basic idea is that the constraint reduces your control over the LOD quality level, often by too much. We must give up the one-level-neighbor constraint, which means the combinatorics between neighboring detail levels can grow much larger: we must build seams that tie together chunks that are two or three levels apart.

When I first thought about how to program this system, I envisioned an octree containing all the chunks. To build a seam between some high-resolution blocks and their low-resolution neighbor, I would collapse one level of the octree in the appropriate place, cross-reference the high-resolution seams to build a new seam, and record that result. This process can be applied repeatedly until we exhaust all the combinations, but programming all this is still a headache. (First we need to collapse portions of the octree by one level, choosing them one at a time, then two at a time, then three at a time, and so on; then we need to collapse one portion of the octree by two levels, but the rest of them by only one level, repeating all the previous combi-

JONATHAN BLOW | *Jonathan is a computer games consultant hanging out in Austin. He can be contacted at jon@number-none.com.*

Hatch something even you'll be afraid of.



Gamebryo™
Beyond NetImmerse

GAMEBRYO IS THE 3D GRAPHICS ENGINE AND TOOLKIT THAT REMOVES THE BARRIERS BETWEEN YOUR IMAGINATION AND YOUR GAME.

ndl

WWW.GAMEBRYO.COM

Gamebryo builds on NetImmerse technology used in these blockbuster games:

Dark Age of Camelot

The Elder Scrolls III MORROWIND

FREEDOM FORCE

Dark Age of Camelot courtesy of Mythic Games. *The Elder Scrolls III: Morrowind* courtesy of Bethesda Softworks. *Freedom Force* courtesy of Irrational Games.

009889

Nothing
ever really be afraid of.

GAMEBRYO™
Beyond Netimmerse

GAMEBRYO IS THE 3D GRAPHICS FRAMEWORK THAT REMOVES
THE BARRIERS BETWEEN YOUR IMAGINATION AND YOUR GAME.

WWW.GAMEBRYO.COM

Gamebryo builds on Netimmerse technology used in these blockbuster games:

Dark Age of Camelot · The Elder Scrolls III: Morrowind · Freedom Force · Emperor: Battle for Dune

DHTML 08 88 8 8 8

nations; then we need to collapse two portions by two levels, and so on. It just feels nasty, and it would require a lot of the unhappy neighbor-navigation code mentioned previously.)

I was unsatisfied with this solution. I wanted a way to deal with all these combinations that was easy to program and easy to understand, so I could put this LOD manager in the core of my rendering system and have some confidence that it actually works.

Triangle-Centricity

Happily, I came up with a simpler system to accomplish all my goals. Two main observations helped me find the simplifications, both of which came about when I decided to stop thinking about octrees, chunk borders, and seams, and moved to an entirely triangle-centric viewpoint.

First, I realized that any triangle, having only three vertices, can touch at most three chunks simultaneously. Thus, if we ever do anything that cares about the combinatorics of more than three chunks at once, whether at preprocess time or at run time, we're complicating the situation needlessly.

The second observation was that, when remapping the high-resolution seams, we actually don't need much information about which chunks neighbor which others. We only need to know which chunks contribute their geometry to which lower-resolution chunks; then we can use that information to rewrite existing seam triangles, and we get all the combinatorics for free.

A Database of Seams

We can think of each triangle as being a triplet of "vertex specifiers": each specifier tells us which chunk of geometry, and which index inside that chunk, represents the vertex we want. Suppose we have some chunk named A. The vertex specifier for chunk A, index 7 can be written as "A7." A seam triangle connecting chunks A and B might be written as (A7, B10, A8).

Suppose we detail-reduce chunk B into a new chunk C, and vertex 10 of B becomes vertex 3 of C. To help create the corresponding seam, we want to rewrite the above triangle so that it becomes (A7, C3, A8). As long as we perform this step properly for every triangle that contains B in a vertex specifier, we will successfully preserve the manifold. It doesn't matter who the neighbors of A, B, and C happen to be. The fact that seams always tie neighbors together becomes an inductive property, caused by the fact that we only made seams between high-resolution neighbors to begin with. We don't need to worry about maintaining this property between resolutions, because it propagates automatically.

At preprocess time, I maintain a database of all existing seam triangles. First, I split input geometry into chunks and put the resulting high-resolution seams into this database. Then I perform the detail reductions and, for each reduction, execute a rewrite rule on the database. The rewrite rule just searches for all triangles containing a certain chunk in their specifiers, writes new versions of those triangles with the new chunks and indices (such as the B10-to-C3 conversion just mentioned), and adds the new triangle to the database. We repeat this process, always adding new triangles to the database, never removing any.

By the time we've reduced our input geometry to a single low-resolution chunk, the database has computed for us all combinations of all seams between neighbors of all possible resolutions. (To get a feel for this, try it with a simple case with pencil and paper.)

We may not wish to store all these combinations, so we can impose limits. For example, we can tell the database never to create seams between chunks that differ by more than two or three levels of detail. We can even set this limit on a chunk-by-chunk basis, with those decisions arising from an analysis of the input geometry.

I have spoken here of manipulating individual triangles, but to reduce memory and CPU costs in the implementation, I group the triangles into seams as before, with the grouping based on the chunk part of their vertex specifiers. So the "chunk membership" is stored in an array on the seam and used for all triangles within the seam; only the vertex indices are stored per-triangle.

All this makes the preprocessing solution rather tidy. But how do we organize these seams so they can be found quickly at run time? The high-level answer to this is that we just store the seam database wholesale and reload it for run-time use. To draw seams between all the solid-rendered chunks on the screen, we should first make an array of those chunks (which we have already done so that we could render them), and then just tell the database, "Give me all the seams that only touch blocks in this set." Then we render those seams. Simple, easy, done.

Now, "database" is a scary word for engine programmers trying to do fast graphics. One might have nightmarish visions of SQL queries happening inside the render loop. In actuality, because we only need one query at run time, we can set up specialized data structures that help us answer that query quickly; our "database" becomes some arrays or hash tables. But to

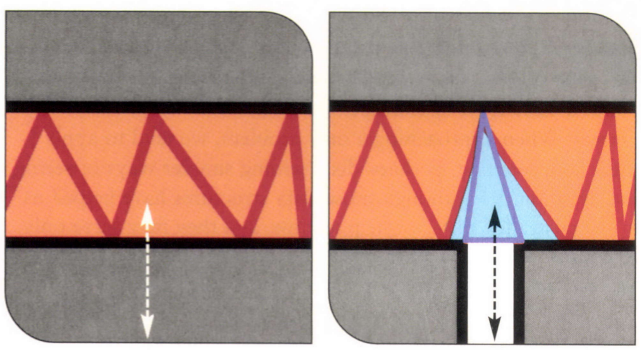


FIGURE 2A (left). Two neighboring chunks, with a seam connecting them (red). We are about to split the lower chunk by clipping it against a plane (green). FIGURE 2B (right). To split the seam, we probably need to subdivide one of the triangles into two (cyan). Once we have two seams, we need to insert a pinhole-filling triangle that connects all three blocks (blue).

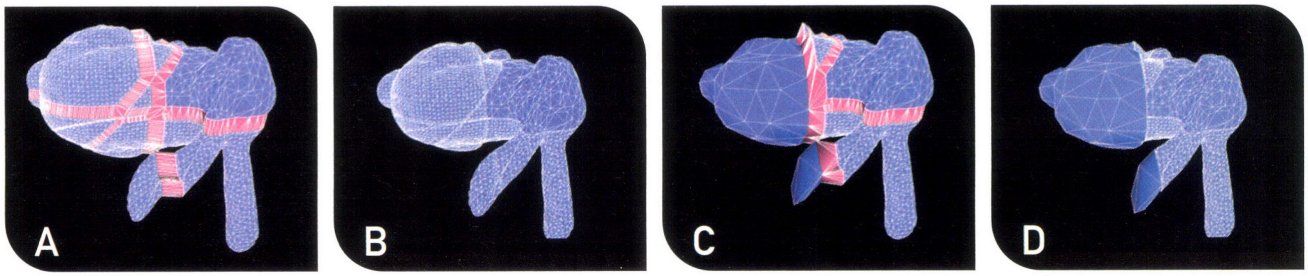


FIGURE 3A. The Stanford Bunny, chopped into arbitrary chunks by four different splitting planes. The original mesh contained 16,000 triangles, but the portion in the upper-right has been reduced by one level of detail. The seam-filling triangles have been drawn in red, and the pieces of the bunny have been pulled apart so that you can see the seams. FIGURE 3B. The same geometry as 3a, but rendered more like it would be in an actual game, without the pieces pulled apart. FIGURE 3C. Like 3a, but now we have reduced the left half of the mesh by three levels of detail. Note that the seams are still properly filled. FIGURE 3D. Like 3c, but without the pieces pulled apart.

maintain simplicity, consider this a problem of "accelerate a database query about vertex specifiers," so try not to fall into the mentality of "store seams in arrays based on neighbors and resolutions," as we did with the height-field renderer.

In this month's sample code, the seams are stored in the database in a single linear array. I performed the database query as follows: First, I mark all the chunks that are being rendered. Then I iterate over every seam in the database and check the chunks it touches (of which, remember, there can be no more than three). If all the chunks are marked, I render the seam, otherwise I don't. After this is done, I unmark all the marked chunks.

This algorithm is $O(m)$, where m is the total number of seams in the database. That's fast enough for the data sizes we are dealing with now, but in a big system it might be a bit slow. By storing seams in arrays on each chunk (which any seam being referenced by multiple arrays), we can reduce the running time to $O(n)$, where n is the actual number of seams you need to draw. Since the task of rendering the seams is itself $O(n)$, it wouldn't help us greatly to try to drive the running time lower. Perhaps I will implement this version of the query next month.

The Moral of the Story

There's a moral to this database story that I would like to pass on. As engine programmers, we're used to thinking about a certain set of concepts and data structures, such as octrees. When approaching a new problem, we tend to apply these concepts first, perhaps disregarding simpler ways of seeing the situation. Even though those data structures have helped us in the past, they may not help us now, and they may serve only to confuse matters. I am reminded of that old proverb "When all you have is a hammer, everything looks like a nail."

Increased Freedom

Now that we use this database rewrite system, neither the preprocess nor the rendering requires an octree. In fact they require very little in the way of data structures. We need only a set of hierarchical bounding volumes for frustum culling and some LOD metric that we can apply to each chunk. That's an amazing amount of freedom, much more than I envisioned when I started this project. That freedom is good; it means anyone using the algorithm will not face many restrictions in how this system must interact with the rest of the renderer.

In fact, nothing in this entire algorithm even cares about the dimensionality of the space we are working in. So if you are some kind of weird physicist running simulations in 11 dimensions and you need a system to perform LOD, maybe this will suit you.

Given all this newfound freedom, I'm going to try something different from what I originally planned. Instead of using a 3D grid of blocks to store the seam, I will employ a system of BSP-style cutting planes, situated at arbitrary orientations. I will then compute these cutting planes based on the density of the scene.

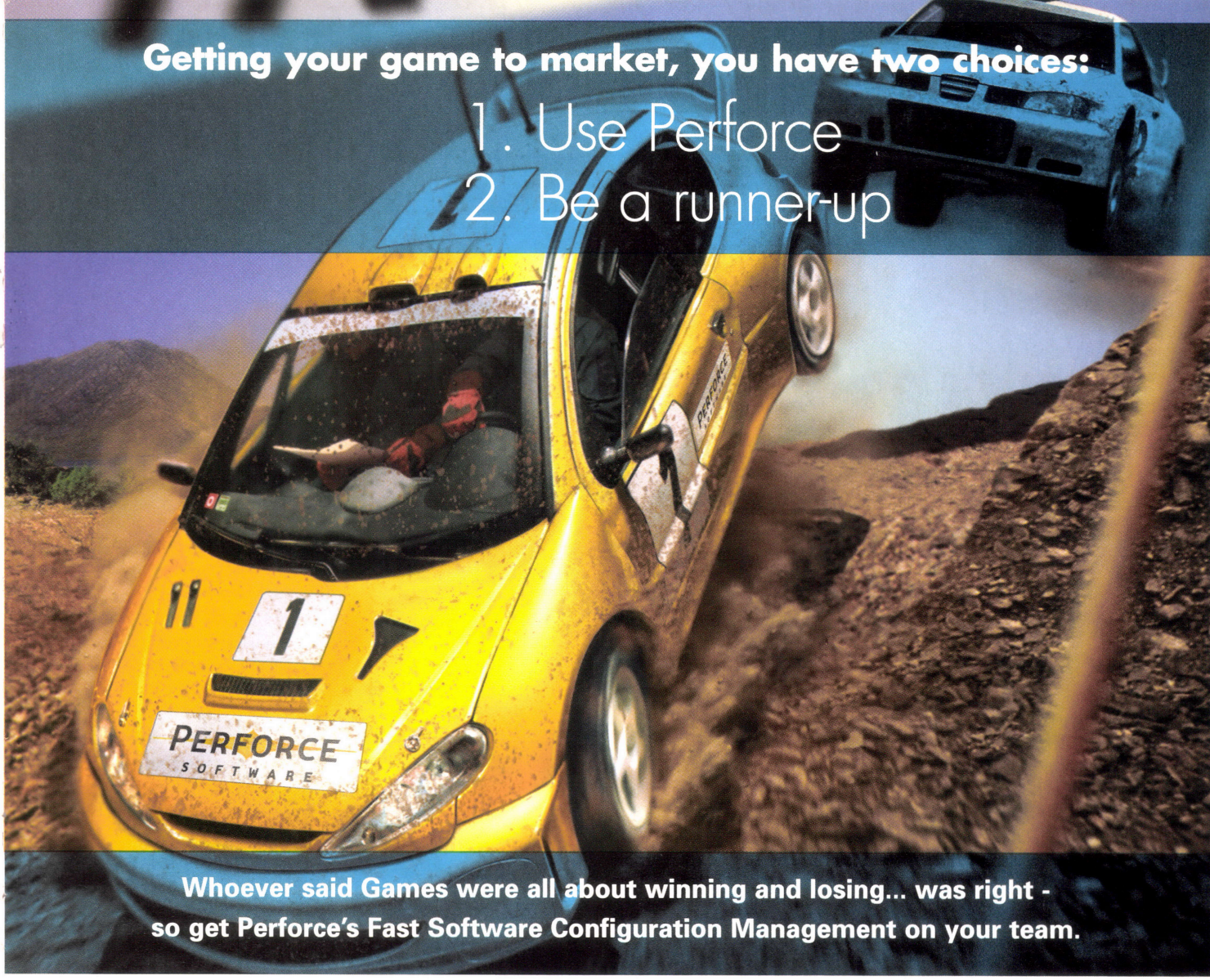
Filling Pinholes

The seam database approach worked so well for LOD generation that I used it for the initial chunk generation as well. I split a chunk into subchunks by applying a single splitting plane and rewriting the seams in the database. Often this will split a seam into two, adding also a single-triangle pinhole-filling seam, as seen in Figure 2. This correctly preserves the manifold for one split, and thus, since we perform one split at a time, it inductively preserves the manifold until we've got all our chunks and are ready to build LODs.

This month's sample code, which you can download from the *Game Developer* web site at www.gdmag.com, contains two different running systems. One of them is the height-field renderer, modified to use the seam database approach. This system serves as a relatively simple introduction to the database, as it doesn't need any of the chunk splitting mentioned above (a height field can be chunked just by applying a window to the array of data).

The second system in the sample code is a new version of the bunny-chopping program, modified now to use an arbitrary number of splitting planes. This program illustrates the BSP-style cutting planes I am talking about, and it serves to verify that the mesh-chunking and pinhole-filling schemes work properly. You can see the results in Figure 3.

Next month we'll look at LOD metrics and discuss methods of choosing splitting planes.



Perforce's Software Configuration Management System is the choice of winners, because no other tool can match the speed, control and reliability that it brings to the management of digital assets (binary and text files).

Maintain your top speed no matter how many users or files. At the core of Perforce lies a relational database with well-keyed tables, so simple operations can be accomplished in near-zero time. Larger operations (like labeling a release and branching) are translated into keyed data access, giving Perforce the scalability that big projects require.

Install it fast, learn it fast, execute operations fast. With other SCM systems, developers face an unpleasant choice: do it the right way or do it the fast way. Perforce's speed and reliability mean the fast way is always the right way.

Work anywhere. Perforce is efficient over high-latency networks such as WANs, the Internet and even low-speed dial-up connections. Requiring only TCP/IP, Perforce makes use of a well-tuned streaming message protocol for synchronising client workspace with server repository contents.

Develop and maintain multiple codelines. Perforce Inter-File Branching™ lets you merge new features and apply fixes between codelines. Smart metadata keeps track of your evolving projects even while they develop in parallel.

Supports all development platforms. Perforce runs on more than 50 operating systems, including all flavours of Windows, UNIX, Linux, Mac OS X and more.

Integrate with leading IDEs and defect trackers: Visual Studio.NET®, Visual C++®, Visual Basic®, JBuilder®, CodeWarrior®, TeamTrack®, Bugzilla™, ControlCenter®, DevTrack® packages, and more.

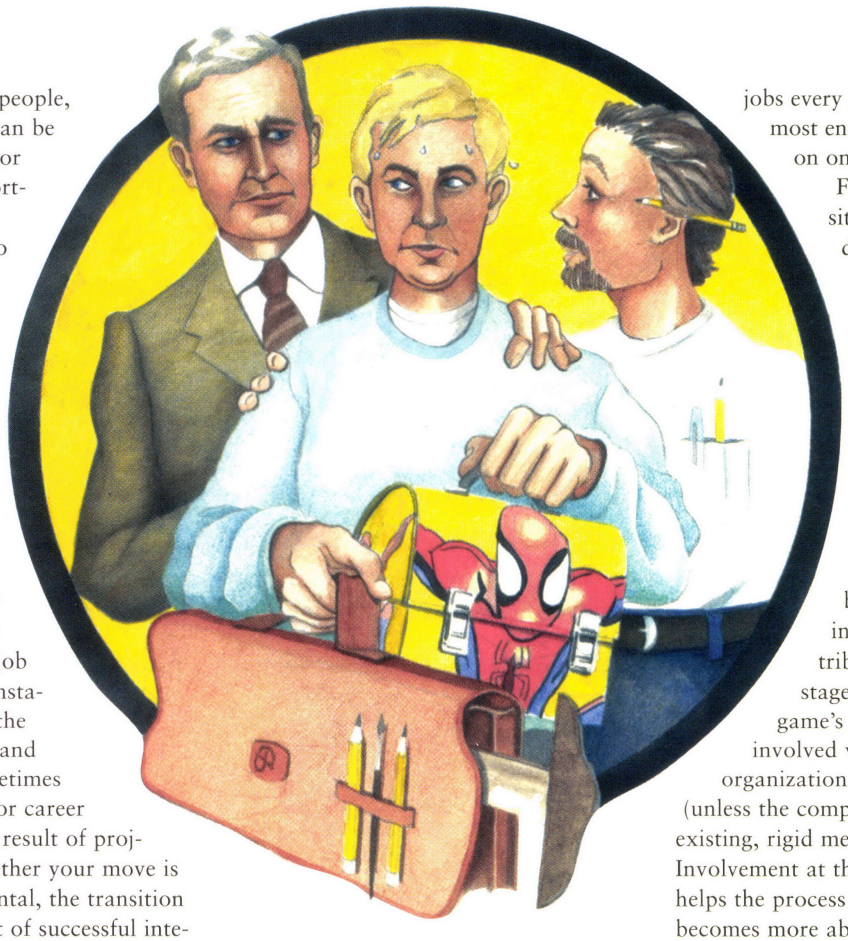
PERFORCE
SOFTWARE

Fast Software Configuration Management www.perforce.com

Download your free 2-user, non-expiring, full-featured copy now from www.perforce.com
Free (and friendly) technical support is on hand to answer any and all evaluation questions.

All trademarks used herein are either the trademarks or registered trademarks of their respective owners.

Transition



For most people, change can be difficult or uncomfortable. Unfamiliarity leads to insecurity, and no one likes to feel insecure. New environments and new experiences can be exciting, but those first steps into the unknown are usually accompanied by some degree of apprehension.

The game industry at present is, like the larger economy and job market, somewhat unstable. Job security for the most part is limited, and changing jobs is sometimes inevitable, whether for career advancement or as a result of project termination. Whether your move is local or transcontinental, the transition process is a vital part of successful integration into a new environment, and it is a crucial time for both individual artists and the companies that employ them if the best long-term results are to be achieved.

The following is a collection of advice gathered from a variety of people within the industry, from relative newcomers to company owners, on how to make the transition process as smooth and effective as possible. The first section addresses the experience of the employee, the second looks at the situation from the employer's side.

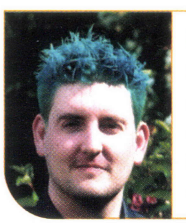
Illustration by Tris Nerima

The New Employee

In almost every case, the newly arrived employee has more to lose than the company employing them if things don't work out. For the employee, changing jobs every few months isn't the most encouraging thing to have on one's résumé.

For artists making transitions, it's important to distinguish between joining a new company to start a project, or at least arriving when the game is still in its very early stages, and joining a company to work on a game that is nearing completion.

An artist joining a project at or near the beginning is likely to be in a position to contribute to the concept stage, helping shape the game's look, and to also be involved with the setup and organization of the art pipeline (unless the company already has an existing, rigid methodology in place). Involvement at this early stage greatly helps the process of transition, as work becomes more about helping to build initial concept and structure than simply fitting into a preexisting slot. Especially with artists' creative nature, the more restrictions that are put on that creativity, the more difficult and less enjoyable the job can be. Chances are, however,



HAYDEN DUVALL | *Hayden started work in 1987, creating airbrushed artwork for the games industry. Over the next eight years, Hayden continued as a freelance artist and lectured in psychology at Perth College in Scotland. Hayden now lives with his wife, Leah, and their four children in Garland, Tex., where he works as an artist at 3D Realms. Contact Hayden at haydend@3drealms.com.*

SecuROM™ Copy Control

IT'S A TECHNICAL KNOCKOUT.

When it comes to game formats, DVD is a real heavyweight.
So you need heavyweight protection for your DVD-based titles.

Already the fastest-growing copy control solution for CD-ROM, **SecuROM** is now the first system to step into the ring to fight DVD-ROM piracy.

With a powerful combination of strong protection, easy encryption, and unmatched compatibility, **SecuROM** is truly a technical knockout.

SecuROM™
get maximum control

Visit us at E3, booth #6245 in Kentia Hall.

For more information or to contact a SecuROM sales representative near you, please visit us at www.securom.com

that a change of jobs will not always bring you in at the start of a project.

Joining a game later on is more difficult, as the artist's contribution is more about content generation and the inevitable reworking of existing assets that characterizes the end of game projects. Stepping into someone else's shoes (who left the vacancy you arrived to fill) or filling a position that has been created at this late stage to make deadlines more achievable puts more pressure on the newcomer, who has to adjust to a new work environment while having to fit seamlessly with an existing art style.

Joining a project in the role of an art director, art lead, or some other senior role, brings additional pressures due to the challenge of leadership. Establishing credibility and respect as an artist is central to a rewarding work experience. This is even more crucial if you join a company in a leadership role. There are several ways in which artists and managers can improve their chances of success, which I've compiled from advice from those that have experienced both good and bad transition periods.

Enthusiasm and Attitude

As the new kid, showing enthusiasm about your new company and in particular the game you are working on is vital. This kind of attitude reassures your employers that they made a good choice in hiring you, something that they might be nervous about until you prove yourself. It's also a form of positive feedback for your coworkers, indicating that you like their work and are happy to be part of what they are creating.

As with most things, it's possible to overstep the mark. A new employee is not best advised to charge into a new job with such overwhelming joy that other artists are swamped by your tide of eagerness. When you join a new company, there will be a natural period of acclimation as everyone works out what kind of person you are. When enthusiasm is taken to the extreme, it can be seen as a negative, or interpreted as an attempt to gain favor with the boss, so moderation is advisable.

Respect and Credibility

It's almost impossible for me to say the word "respect" without lapsing into a Marlon Brando parody. And while some game companies may feel like they are operated by the Corleone family, gaining respect for your work as an artist is a legitimate concern.

Respect, however, can be a difficult thing to pin down. In many walks of life, respect is a result of status and a person's achievements. This translates to some extent to the game development industry, and particularly as an artist, a large part of the respect you command comes from the quality of your work. Every artist you work with has his or her particular skills and areas of expertise. It's part of the natural process of interpersonal assessment that your work will be used to inform other artists' opinions of you, and vice versa.

As a professional, you will have to earn the respect of those that you work with through your conduct and attitude. Respect is most effectively built on a mutual basis, and demonstrating that you respect the work of others around you is an important step in this direction.

Acceptance and Change

Being accepted into the team is about more than just the allocation of work and a name on the credits. As with any social grouping, there are unwritten rules about how the group works, who fits into which role, how different personalities interact, and the inevitable degree of company politics to negotiate.

One important way to gain acceptance is to ensure that your priority on arrival is to learn rather than attempting to instigate changes. Whether you're a junior artist or the newly arrived creative director, learning about the team, the game, and how everyone currently operates is certainly the place to start. The fact that you can see improvements that can be made from day one does not necessarily mean that immediately pointing those needs out is the most productive approach to take, even if you are in a position to do so.

It's true that as a group of professionals, everyone should be able to accept suggested changes that will benefit the project, even if they come from the new guy. But human nature cannot be discounted, so a "settling in" period is advisable. This time can also be used to build a more complete picture of your new company and new position. Having a solid feel for these areas will help you suggest changes in the most effective manner.

Conflict and Criticism

While it's best to avoid workplace beatings whenever possible, artists have the difficult job of dealing in an area that is essentially subjective in nature. For programmers, code tends to either work correctly or not. Artists, however, are subject to opinions of a more ethereal nature. Whether you are giving opinions or receiving them, you need to be tactful, measured, and able to accept opinions with which you don't agree.

New artists especially can be subject to a large amount of feedback (some of it negative), while they find their footing. Newcomers must accept this as guidance more than criticism as they learn new stylistic and methodological processes. There is also a tendency for senior members of a team to feel that they need to be seen as having input, which often comes in the form of requested changes. This is hard to counter but will usually diminish noticeably once you have been there for a while. Overall, it's vital to take criticism well regardless of its origin, and to discuss changes rather than argue about them.

First Impressions

Don't arrive at your new job with the intention of putting your stamp of ownership on the game. Every developer I have talked to has expressed dislike of the prima donna attitude sometimes displayed by those who believe they are

Experience Maya® 5

The world's most powerful 3D tool for game development!

CAN YOU IMAGINE®

© 2003 Alias|Wavefront, a division of Silicon Graphics Limited. All rights reserved. The Alias|Wavefront logo and Can You Imagine are registered trademarks and Alias|Wavefront, the Maya logo is a trademark of Alias|Wavefront, a division of Silicon Graphics Limited. Maya is a registered trademark of Silicon Graphics, Inc., used exclusively by Alias|Wavefront, a division of Silicon Graphics Limited. The Game Developers Choice Awards logo is a trademark or registered trademark of the IGDA. Metroid Prime is a trademark of Nintendo of America Inc. Sly Cooper and the Thievius Raccoonus are trademarks of Sony Computer Entertainment America Inc. Kingdom Hearts is a registered trademark of Square Enix USA. All other trademarks, service marks, trade names, or product names are property of their respective owners.

Advances in rendering technology and workflow offer unique level and character lighting possibilities. Numerous fine-tuned features increase the productivity of game artists and programmers. New features and functionality throughout Maya 5 give game developers more creative resources.

Alias|Wavefront™ congratulates all the nominees in this year's Game Developers Choice Awards™, with a special acknowledgement to those winners who used Maya to create their award-winning titles:

Game of the Year
Metroid Prime™
Developer: Retro Studios / Publisher: Nintendo of America

Rookie Studio of the Year
Retro Studios for Metroid Prime™

Excellence in Level Design
Metroid Team for Level Design in Metroid Prime™
Developer: Retro Studios / Publisher: Nintendo of America

game developers CHOICE awards

Original Game Character of the Year
Sly Cooper from Sly Cooper and the Thievius Raccoonus™
Developer: Sucker Punch
Publisher: Sony Computer Entertainment America

Excellence in Visual Arts
Tetsuya Nomura for Art direction in Kingdom Hearts™
Developer: Squaresoft
Publisher: Sony Computer Entertainment America

Discover the unique game art creation possibilities of Maya 5 at www.aliaswavefront.com/maya or visit Meeting Room 501, upper level, west hall, LACC

Alias|wavefront®
www.aliaswavefront.com

special. Even if you are special, making a game always has to be a team effort. Anyone who works against this, especially if they are new, is unlikely to be appreciated.

Dangerous comparisons. One of the most counterproductive things a new employee can do is to constantly harp back to a previous job, especially when the comparisons are unfavorable to one's present employer. Statements like "At my last place that was all taken care of in the editor, which was so much better than what you have here" don't help, and referring to your present employer as "you" rather than "we" is never a good thing.

Mouthing off. Our industry may be large in terms of revenue, but it's surprisingly small and incestuous in terms of the workforce. With this in mind, it's never a good idea to bad-mouth people you may have worked with previously, chances are that you'll be working with someone who knows them and may well not share your opinion. This is not to say that you need to be the epitome of sweetness and light, it's simply a case of using discretion when telling your story about "John and the transsexual hooker."

Politics and power struggles. If your new company has a lot of that kind of thing going on, there is no easy way to avoid office politics. If you are an art lead, it's important not to distance yourself from the art team by taking the "executive lounge" route. The best leaders have always led from the front and by example, which is particularly applicable to the collaborative nature of game art. If you are the new artist on the team, it's important that you are not projecting an image of being in competition with the other artists; once this begins to happen, the team loses its coherence in the face of one-upmanship, which always builds resentment.

The Employer's Role

Similar to the points made about the new employee's role, there are several things that an employer needs to consider when dealing with a new employee's transitional period. The following advice has been collected mainly from the experiences of those I have spoken with as they themselves have moved into new jobs over the years. The following illustrates the reciprocal importance of a new employee's transitional period. The new employers also need to help make the transition smooth if they want to get the best out of their new hire.

Be prepared. Preparation is a basic point, but it's often overlooked when developers are deeply occupied in actually making a game. Little is more frustrating for a new employee, ready and raring to go in a new position, than turning up for work and finding that a computer hasn't even been ordered, let alone set up. My informal research suggests over half of the artists beginning work have faced delays ranging from a few days to over a month while waiting for hardware.

Moreover, a top-of-the-line PC is of little use if the none of the software the new artist needs is on it. Add to this the availability of scanners, digital cameras, graphics tablets, and basic furniture needs, and it becomes clear that employers need to be on the case before the artist steps through the doors, ready to work. Not only do delays like this waste the artist's and the company's time, but first impressions matter for employers too.

Beyond software and hardware needs, an employer needs to have informed the relevant people (ideally the whole company) of any new arrival so that there is a plan in place ready to streamline the integration process. Sitting in front of a screen, waiting for someone to figure out what to do with you is not the best introduction to your new job, but unfortunately, it happens all too often. Because there are no hard lines between what's right and what's wrong in art, it's even more important that a new artist be given the necessary information and guidance to learn what he needs to. Details such as acceptable polygon count will be dependent on the specifics of the game, its engine, and the platforms it's running on, and should be communicated as soon as possible.

Be open and inclusive. Sometimes game companies (especially larger ones) treat a new employee as if they are some kind of intruder. Secrecy is usually a tool of managerial manipulation and an indication of a lack of trust. While that's another topic entirely, employers need to be as open and honest as possible if they expect their new employee to trust them and be honest in return. I am not talking about disclosure of the owner's salary, but things that need to be made clear are important issues, such as an explanation of the real chain of command. Every company, even those that deny it, have some form of hierarchy; the more covertly this system operates, the harder it is for a new employee to understand who reports to whom and more importantly, who actually has the final word. Pretending that person A is the one to sign off on your new character design is a pointless exercise if in reality, his or her superior can decide it's unacceptable later that week.

Employers must also remember to include a new employee as much as possible. Not knowing the routine can isolate new staff, so the employer should actively attempt to involve them in as much as possible both during and after work if necessary (for example, if they are new to the area).

Transition Is Short, Success Is Long

While the sensitive artist stereotype is largely fictitious, and no one that I know would burst into tears if they found it difficult to transition into a new job (well, maybe I can think of one person who might), both employer and employee can learn from the experiences of others to make a transition period more successful. Hopefully the considerations presented here will make your next transition, whether from the employer's or the employee's perspective, more profitable in the long run by helping avoid common mistakes and frustrations that often get things off to a bad start.

Publish and perish?

Your game is going to be a massive hit.
"Anticipating 200 thousand subscribers"
flows from your lips like honey.

Great. Let's talk logistics.
Those 200 thousand players are going
to require some back-end heavy metal that
dwarfs the Colossus of Rhodes.
A second mortgage is no big deal right?

No big deal if you like your in-laws...
because that's where you'll be living if only
200 Macedonians show to battle the Greeks.

There is another way.
The Butterfly Grid is a complete software solution
that has everything you need to develop, test, operate, manage
and support real-time, immersive online games.

Using Butterfly Grid you might start out with the small battles,
but when a million players show up you win the war...
instant infrastructure stands ready for when you need it.
But doesn't cost you if you don't.

So start small, get big. Go Grid or go home.

Contact us at 304-260-9520, extension 9 for more details.
Or visit our web site, www.butterfly.net.

Butterfly.net™

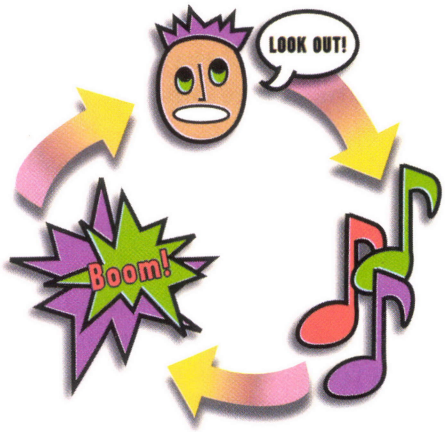
Go Grid

Managing the Nonlinear Mix

Game audio elements are typically created and mastered independently of one another. Within a game, any number of these independent sound effects, voice tracks, and music tracks are played simultaneously. In a situation where the order and layering of these elements is not predetermined but rather determined by player interaction, a nonlinear mix occurs. In this situation, problems may arise causing balance issues between sound effects, music, and dialogue.

Furthermore, unforeseen combinations of sounds that are dominant in similar areas of the frequency spectrum can create a mix that sounds "muddy" or "cluttered." While the actions of the game player determine the timing of certain audio events — and in many cases the frequency at which those events occur — the sound designer has control over each element's amplitude, its spectral content, and the texture or timbre of the overall game soundtrack. With some forethought given to texture, spectral, and level management, sound designers can reduce the common problems that result from a nonlinear mix.

Texture management. Texture management is the organization of the audio assets. By examining the overall style and audio requirements for the game and subsequently its individual levels or subsections, it's possible to determine the game's overall sonic texture. Some questions to consider include: Should the ambient sound effects dominate, or is the gameplay music-driven? What role will spoken dialogue play, and what is the priority of that role? Will triggered, event-driven sounds interrupt the dialogue or disrupt the music tracks? The answers to these types of questions will help determine the overall texture of the soundtrack. Based on these answers you can focus the audio design for each segment of the game to highlight those key audio elements or textures deemed most important.



Level management. Level management refers to the amplitude of individual sonic elements or classes of sonic elements within the overall soundtrack of the game. Fundamentally, level management describes the volume levels of the sound effects, music, and dialogue, as well as the overall dynamic range of the game's soundtrack (that is, the difference between the loudest and softest sounds in the game). More in-depth level management includes looking at the amplitudes of sound element groups. For example, how loud are the footsteps in relation to the ambient sound effects in relation to an NPCs' sound effects in relation to the NPCs' dialogue? So while texture management determines what sounds should be occurring on a per-level basis, and the player determines when those sounds are playing, level management establishes relative amplitudes for those sounds.

The implementation of level management occurs at several stages within the game development process. The first set of amplitude decisions are made when the sound elements are mastered as individual files. Level management occurs again when the sounds are integrated into the game during the audio coding process. Playback levels are set to be executed by the audio engine during gameplay. Finally, some game titles delegate a portion of audio level management to the game player, giving the player overall balance control between voice, music, and sound effects from a sound options menu.

Spectral management. Spectral management relates to the allocation or consideration of the spectral content of sounds or groups of sonic elements within the game. Some nonlinear mix problems can be reduced or eliminated through the use of creative equalization or pre-allocated frequency bandwidths on specific audio elements. For example, in areas with constant music, heavy ambient sound effects, or both, an equalization curve could be applied to those elements allowing more room in the spectrum for additional event-based sound effects or dialogue. Attenuation in the lower-mid-range frequencies could allow for additional intelligibility in the dialogue track. Another notch in the upper end of the spectrum could help accommodate transient sound effects that might be triggered.

While the concepts of texture, level, and spectral management are not new to the audio mix environment, careful attention must be paid to these concepts in the early stages of sound design. Early and thorough planning will reduce later headaches and result in a more cohesive and well balanced nonlinear mix.

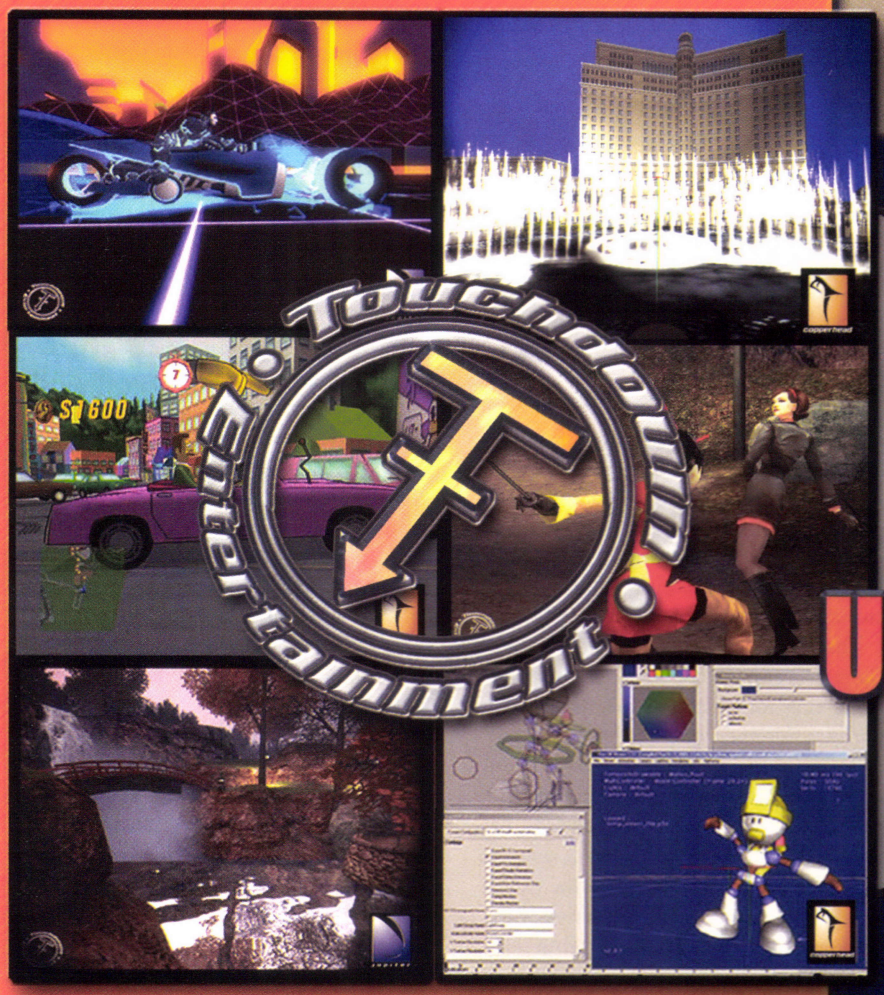
MIKE VERRETTE | *Mike is the audio director for Wicked Noise and member of the Game Audio Network Guild. When not managing his spectrum he can be reached at mike@wickednoise.com.*

Being

OLD SCHOOL

isn't

always good.



time to

UPGRADE.

www.touchdownentertainment.com

copperhead

PlayStation.2 • XBOX • NINTENDO GAMECUBE. • PC

jupiter

©2003 Touchdown Entertainment, Inc. Copperhead and the Copperhead logo are trademarks of Touchdown Entertainment, Inc.

The Hobgoblin of Little Minds

A minor trivia game for your enjoyment. What is "the hobgoblin of little minds"? Two points for the correct answer. Score another three points if you know who said it. And if you know the entire quotation, give yourself a big five points.

The answer?

"A foolish consistency is the hobgoblin of little minds, adored by little statesmen and philosophers and divines." — Ralph Waldo Emerson

I only scored the first two points myself. When I looked up the full quotation, I was surprised to see that Emerson was insightful enough to condemn not consistency, but a foolish consistency.

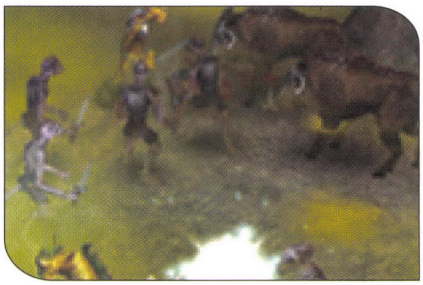
That's trumping information. Be consistent, but don't be foolishly consistent.

Is that one of the 400 Rules of game design? I think it's too vague. Foolish is as foolish does. So I'm turning to you for help in clarifying it. I've been admonished for taking the obvious direction with my choice of the rules I've published here, and I think rightfully so. I started with the easier ones that were hard to argue with, thinking that was a good way to prime the pump with rules. But now that there's a steady flow, it's time to risk a little more controversy.

It's obvious that some consistency is a good thing. You wouldn't want a game that showed every enemy unit in red and every friendly unit in green — except for a friendly medic unit with a big red cross that you blew up the first time you saw it approaching. And consistency for control interfaces is important too, you wouldn't want one part of the game to have one interface and then suddenly change to a different one.

Or would you?

If you have a platform game and you're happily running and jumping but then get the magic hat that lets you fly, don't you need to change the interface, at least a little? If you switch from a first-person view to a 3D strategic map view, you have to change the way the player selects a target — don't you?



NEVERWINTER NIGHTS may not always be consistent, but it does have lots of hobgoblins!

So perhaps the rule is "Be consistent in interface as long as the context of what the player controls remains constant."

And yet if you can remain consistent even when the context of control has changed — perhaps by turning that jump button into a flap button — isn't that even better?

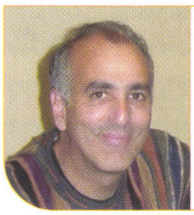
Let's look at it from a slightly different perspective. One reader suggested to me that a good rule would be "Be consistent in player feedback." If smashing a crate (to take an original example) gives you something useful, don't have the 200th crate you find blow up in your face and take away a life.

That seems obvious. But at the recent Game Developers Conference, I had an interesting conversation with Mark Cerny, a designer who has been a design contributor to games totaling over $1 billion in sales (that's billion with a *b*). Regarding that very issue of rewards, he told me that he thinks that every once in a great while you should blow the player up for no reason. My first reaction was, "He's lost his marbles!" But having reason to believe Mark is quite aware of where his marbles are, I started to think about the player's experience.

Let's say the player is an hour or two into a game and has opened 200 crates, all with good stuff in them. The initial thrill has probably long since worn off, and opening crates has become rote, even boring. Then, with no warning, the next crate blows up in the player's face, costing a life. What's the player going to think? The player won't stop opening crates — after all, 200 had rewards in them. But the player will approach every subsequent crate with more care and more excitement. When crate number 600 blows up, the player may even think there's a pattern. But Mark insisted there must be no pattern. I like that — it fits in with the AI rules I mentioned in my January 2003 column ("AI Without Pain") about a little randomness and the power of suggestion. And it suggests that the really interesting and useful guidance to a game designer about consistency is when to break that consistency.

Or to put it another way, when does too much consistency become foolish?

I have some ideas about that, but I'd like to turn to you, the readers. Send me some one-sentence imperative rules that a fellow designer can actually implement, and perhaps some associated simple exceptions. Please refrain from long discourses, I'm hoping to boil these down into a few simple, clear rules and information about when to trump them, and why. I'll discuss the most provocative ones in a future column. Who knows, we might even find one consistent rule about consistency. That would be a good thing — wouldn't it?



NOAH FALSTEIN | *Noah is a 23-year veteran of the game industry. His web site, www.theinspiracy.com, has a description of The 400 Project, the basis for these columns. Also at that site is a list of the game design rules collected so far, and tips on how to use them. You can e-mail Noah at noah@theinspiracy.com.*

A New Design Freedom

3Dlabs
Graphics Evolved
A CREATIVE Company

Introducing the World's First Fully Programmable Visual Processor

The sky's the only limit to your design creativity with Wildcat® VP—an astounding breakthrough in 3D workstation graphics. The industry's first fully programmable Visual Processing Unit (VPU) powers a family of accelerators that deliver unmatched productivity for professional OpenGL® and Direct3D®—based applications. But that's not all. With unrivaled flexibility, Wildcat VP is enabling a new generation of applications to extend the boundaries of interactive visual realism. Experience the ultimate in graphics design freedom today. Be ready for visual programmability tomorrow. Wildcat VP from 3Dlabs.

Leading OpenGL and Direct3D performance	A full family of 64MB and 128MB boards
Over 200GFlop and 1.2 TeraOp VPU	High performance virtual memory
Professional-grade reliability and quality	Dual independent displays

Wildcat VP

www.3dlabs.com

3Dlabs is a registered trademark of 3Dlabs Inc. in the United States and other countries.

melissa federoff

Improving Games with User Testing: Getting Better Data Earlier

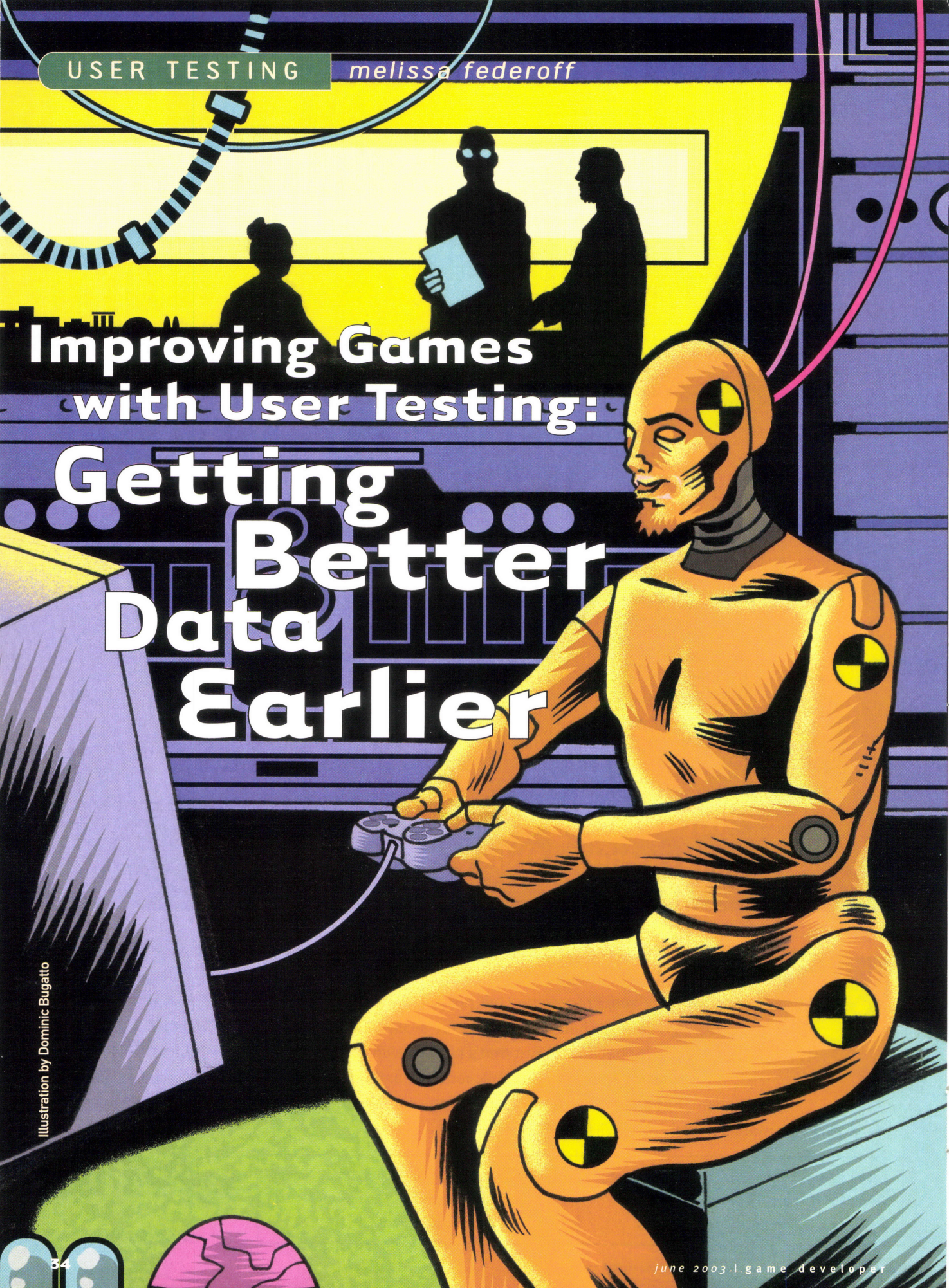


Illustration by Dominic Bugatto

Most game designers do not acquire major feedback on their products until beta, when quality-assurance representation on a given product increases. QA testers are trying to break the game: they're finding bugs, they're finding balancing issues, they may even be finding major gameplay problems. But even if they can offer feedback on design problems, the game is meant to ship in a matter of a few short months, so it might be too late to fix most of them.

How can game teams get design feedback earlier, when the game isn't in final form yet? Several possible methods come from a discipline called usability. Central to the concept of usability is the evaluation of users, or target audience. By borrowing concepts from the field of usability and applying them to games, designers can get ongoing data on how to improve their games right from the consumer.

What Is Usability?

Usability is a field of study where a product is tested on actual users for efficiency, effectiveness, and satisfaction throughout development. These three measures help to determine if the design goals of a product are being met. In the case of productivity software, testing these elements of usability helps to ensure that the user can meet the product's goals, that the user can achieve them in a way that expends the least amount of energy, and that the software provides an overall satisfactory experience to the user. Often in productivity software, efficiency and effectiveness are the focus of testing, because these are the main selling points for businesses, the major buyers of the products.

How Usability Applies to Games

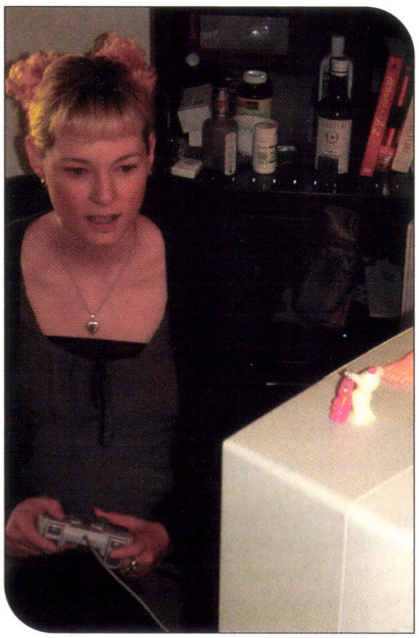
With games, we have an entirely different situation from productivity software. By selling to individuals rather than businesses, there is an incredible amount of competition, both from the vast amount of games on the market and from all the other types of entertainment consumers can choose to buy and engage in at any given time. We are also selling an experience, not a tool. We want players to be able to interact with the game's interface efficiently and effectively, and we want them to play through levels as close to our ideal path as possible. Above all else, we want them to have fun.

Using usability terms, then, we can and should be testing for effectiveness and efficiency, but for games we want to focus our efforts on measuring satisfaction. These usability measures can help us to judge whether players are able to use the controller adequately, check their status in the game correctly, use menus without frustration, make their way through levels successfully, learn skills as they progress, be challenged appropriately, and have a desire to keep playing because they are experiencing the type of entertainment and amount of fun they expect.

Test Early, Test Often

Game designers are no different from other types of designers. They generally think they can judge the quality of their own project and anticipate how their audience will perceive it. Though sometimes it is possible to guess how someone else will interact with a product, design choices can only be verified through testing.

A full-scale production is too large of an investment to base on hunches or artistic desires. Commercial game development is not creating art for art's sake; it's creating a product to sell to a consumer. Therefore, developers need to test, at every stage of development, whether the product is reaching its goals.



Volunteer participant Lily Childs plays RTX Red Rock during an in-house test at LucasArts.

Usability vs. Quality Assurance

In order to assess whether a design is working well, it needs to be tested on the target audience — those who are likely to buy and use it after it is released. QA testers are not representative of the target audience of a game, because they are professional game players. The vast majority of game consumers do not get the level of exposure to games that a QA tester has. Therefore, testers are unlikely to interact with a game in the way a nonprofessional player would. Also, since QA testers are often a part of the actual game team, they, like designers, are often too close to a game to judge it in an objective way.

MELISSA FEDEROFF | *Melissa works for LucasArts Entertainment Company. She began her study of the usability of games as a graduate student in the MIME (Master's in Immersive Mediated Environments) program at Indiana University. Her thesis, which is an expanded version of this article, can be found at www.melissafederoff.com/thesis.html.*

Getting accurate impressions of a design is critical, and so is getting the feedback in a timely way. Acquiring information about a design late in development can mean one of two things: either the necessary changes are not made and the product is not improved, or changes are made and the product is derailed from its production cycle. If feedback is acquired about design decisions as they are being made, before large investments of time are made in them, the product will more easily be improved.

Implementing Usability Techniques

Usability offers the game development industry different methods from which to choose for evaluating games. These methods vary in the type and quality of data they produce based on the resources available to implement them.

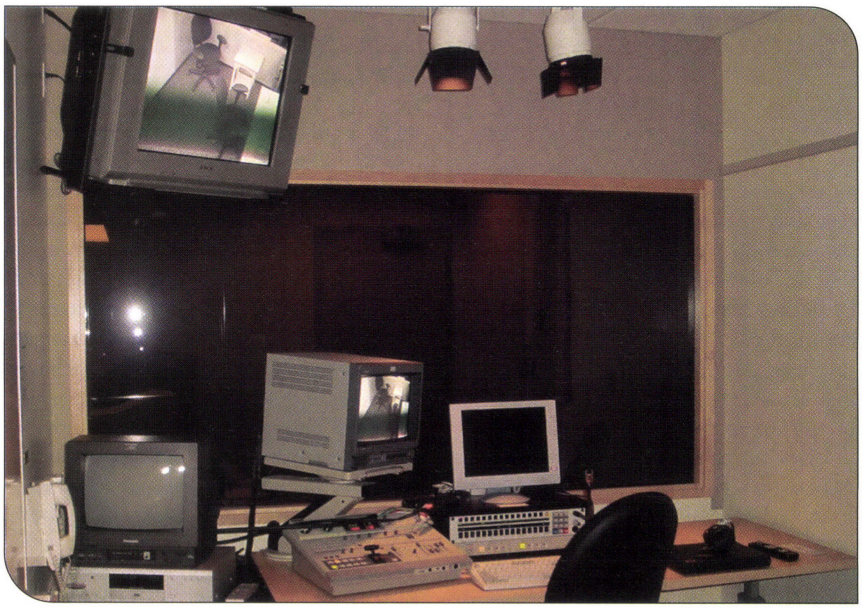
Some methods — such as expert evaluations — do not require the cost of labs and participant compensation, but they do require the cost of having a usability professional run the tests. Other methods, like play-testing, can be done in-house to avoid participant compensation and coordination costs, but still require employee time.

No matter what method is chosen, usability testing costs money up-front, but investing in it can save time by avoiding costly mistakes and should increase the overall quality of products if performed correctly.

Beginning Steps

Before engaging in any usability tests, it's best to incorporate extra time into the production cycle for testing and fixes so that changes can be made as problems are found. Doing multiple rounds of testing is important for finding new problems as they arise, as well as for verifying that any changes that have been made are working.

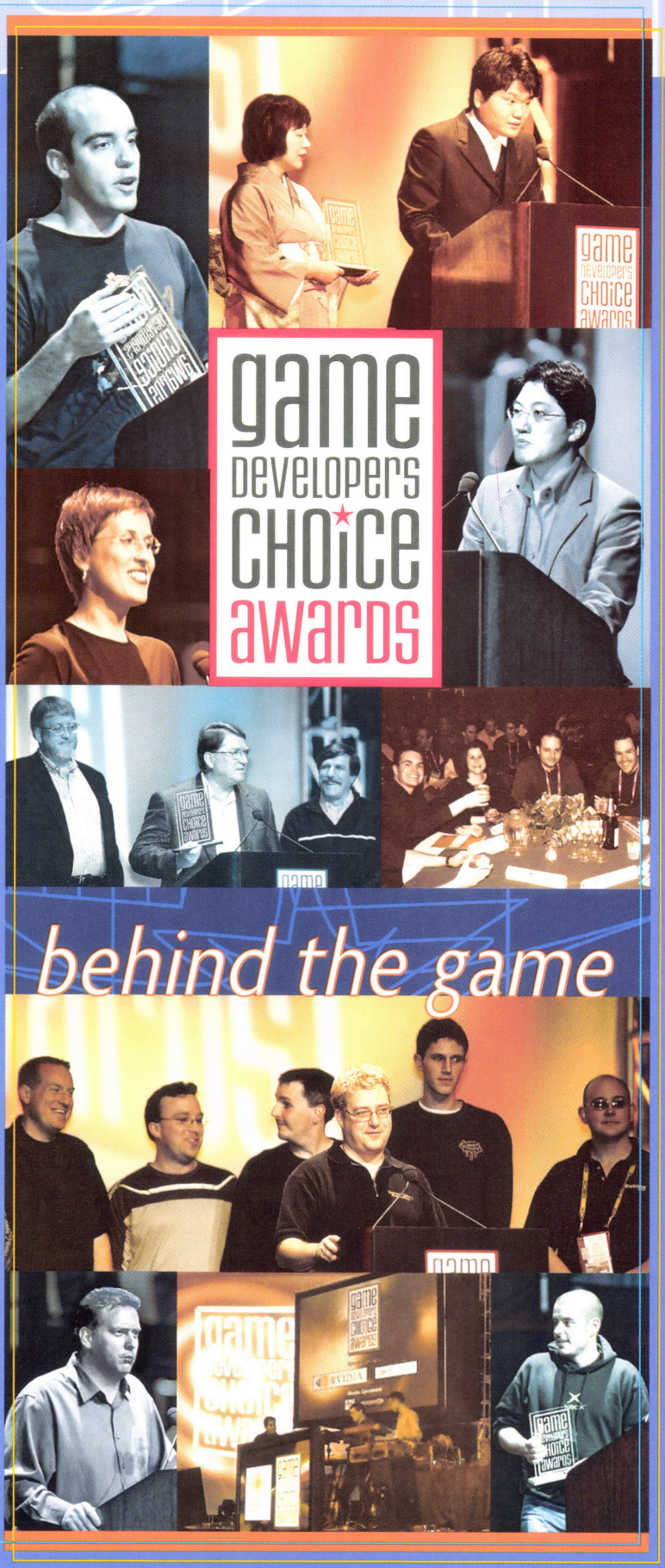
Concept testing. Ideas can be tested even before they hit the digital realm. Reading story scripts or showing concept art to members of the identified target audience can help display what ideas are resonating and what ideas are not getting across the way the designer intended.



Microsoft Game Studios User-Testing Group Usability Lab: observer side.

Microsoft Game Studios User-Testing Group Usability Lab: participant side.

Prototyping. After an idea is concept tested, it can be prototyped. This can be on paper, such as a story flowchart or an interface mock-up, or it can actually involve creating a mini playable version of the game. A playable version could involve designing a level that incorporates all of the important components of



And the Game Developers Choice Award goes to...

igda international game developers association

Game Developers Choice Awards Nominees

GAME OF THE YEAR
- BATTLEFIELD 1942 by Digital Illusions
- GRAND THEFT AUTO: VICE CITY by Rockstar North
- METROID PRIME by Retro Studios
- NEVERWINTER NIGHTS by BioWare Corp.
- TOM CLANCY'S SPLINTER CELL by Ubi Soft Montreal

ORIGINAL GAME CHARACTER OF THE YEAR
- Blinx from BLINX THE TIME SWEEPER
- Ratchet from RATCHET AND CLANK
- Rau from THE MARK OF KRI
- Sam Fisher from TOM CLANCY'S SPLINTER CELL
- Sly Cooper from SLY COOPER AND THE THIEVIUS RACCOONUS

ROOKIE STUDIO OF THE YEAR
- Arkane Studios for ARX FATALIS
- Day 1 Studios for MECHASSAULT
- Gas Powered Games for DUNGEON SIEGE
- Pipeworks Software for GODZILLA: DESTROY ALL MONSTERS MELEE
- Retro Studios for METROID PRIME

EXCELLENCE IN AUDIO
- GTA Team for sound design in GRAND THEFT AUTO: VICE CITY
- Takayuki Kawagoe, Hideki Naganuma and Fumitaka Shibata for sound design in JET SET RADIO FUTURE
- Jack Grillo, Rebecca Hanck, Erik Kraber and Yuan Liu for sound effects in MEDAL OF HONOR: ALLIED ASSAULT
- Metroid Team for sound effects in METROID PRIME
- Andrew Boyd, Chris Hegstrom, Robb Mills and Howard Shore for composition in THE LORD OF THE RINGS: THE TWO TOWERS

EXCELLENCE IN GAME DESIGN
- Romain de Waubert de Genlis and Team for game design in BATTLEFIELD 1942
- GTA Team for game design in GRAND THEFT AUTO: VICE CITY
- Metroid Team for game design in METROID PRIME
- Satoru Iwata, Yoshiaki Koizumi, Shigeru Miyamoto, Takashi Tezuka and Kenta Usui for game design in SUPER MARIO SUNSHINE
- Splinter Cell Team Leaders for game design in TOM CLANCY'S SPLINTER CELL

EXCELLENCE IN LEVEL DESIGN
- Metroid Team for level design in METROID PRIME
- Brian Allgeier, Mark Cerny, Lesley Mathieson and Colin Munson for level design in RATCHET & CLANK
- Brendan McNamara, Chun Wah Kong and the SCEE Design Team for level design in THE GETAWAY
- François Pelland and Team for level design in TOM CLANCY'S SPLINTER CELL
- Pancho Eekels, Dave Ewing and James Schmalz for level design in UNREAL TOURNAMENT 2003

EXCELLENCE IN PROGRAMMING
- Mike Biddlecombe, Scott Bilas, Bartosz Kijanka, James Loe, Eric Tams and Chad Queen for programming in DUNGEON SIEGE
- Metroid Team for programming in METROID PRIME
- Mark Brockington, Scott Greig, Jason Knipe, Don Moar and Don Yakielashek for network programming in NEVERWINTER NIGHTS
- Antoine Dodens and Team for graphics programming in TOM CLANCY'S SPLINTER CELL
- Warcraft III Team for programming in WARCRAFT III: REIGN OF CHAOS

EXCELLENCE IN VISUAL ARTS
- Jonathan Chey, Ken Levine and Team for art direction in FREEDOM FORCE
- Tetsuya Nomura for art direction in KINGDOM HEARTS
- Hokyo Lim, Suzanne Kaufman, Dev Madan, Augie Pagan and Karin Yamagiwa for art direction in SLY COOPER AND THE THIEVIUS RACCOONUS
- Nicolas Cantin, Frédérick Gagné and Benoit Sokal for art direction in SYBERIA
- Jay Beard, Erik Medina, Jeff Merghart, Dan Mueller and Tim Neveu for animation in THE MARK OF KRI

EXCELLENCE IN WRITING
- Denis Dyack and Ken McCulloch for writing in ETERNAL DARKNESS: SANITY'S REQUIEM
- GTA Team for writing in GRAND THEFT AUTO: VICE CITY
- Daniel Vavra for writing in MAFIA: THE CITY OF LOST HEAVEN
- Craig Hubbard and Team for writing in NO ONE LIVES FOREVER 2: A SPY IN H.A.R.M.S. WAY
- Clint Hocking and JT Petty for writing in TOM CLANCY'S SPLINTER CELL

Generous support from our sponsors made the Game Developers Choice Awards program and event possible. It is an honor to work with sponsors who appreciate the need for a developer centric awards program.

Sponsored by:

Media Sponsors:

The International Game Developers Association presents

The Game Developers Choice Awards

Celebrate the talent behind the game

One evening in early March, during the Game Developers Conference, over 2000 of our industry gathered to celebrate the Game Developers Choice Awards. They came to honor programmers, artists, level designers, audio artists, writers, and game designers. Among the select recipients: a raccoon, an AI innovation, a reincarnated classic, arcade renegades, an industry leader and, by proxy, a visionary's widow and son.

The IGDA is proud to highlight here the recipients for the third annual Game Developers Choice Awards. At its heart, the Choice Awards – the only developer nominated, voted and selected awards program – celebrate the talent behind the game . . .

GAME OF THE YEAR
METROID PRIME by Retro Studios

ORIGINAL GAME CHARACTER OF THE YEAR
Sly Cooper from
SLY COOPER AND THE THIEVIUS RACCOONUS

ROOKIE STUDIO OF THE YEAR
Retro Studios for METROID PRIME

EXCELLENCE IN AUDIO
Jack Grillo, Rebecca Hanck, Erik Kraber and Yuan Liu for sound effects in
MEDAL OF HONOR: ALLIED ASSAULT

EXCELLENCE IN GAME DESIGN
Romain de Waubert de Genlis and Team for game design in BATTLEFIELD 1942

EXCELLENCE IN LEVEL DESIGN
Metroid Team for level design in
METROID PRIME

EXCELLENCE IN PROGRAMMING
Mark Brockington, Scott Greig, Jason Knipe, Don Moar and Don Yakielashek for network programming in NEVERWINTER NIGHTS

EXCELLENCE IN VISUAL ARTS
Tetsuya Nomura for art direction in
KINGDOM HEARTS

EXCELLENCE IN WRITING
Clint Hocking and JT Petty for writing in
TOM CLANCY'S SPLINTER CELL

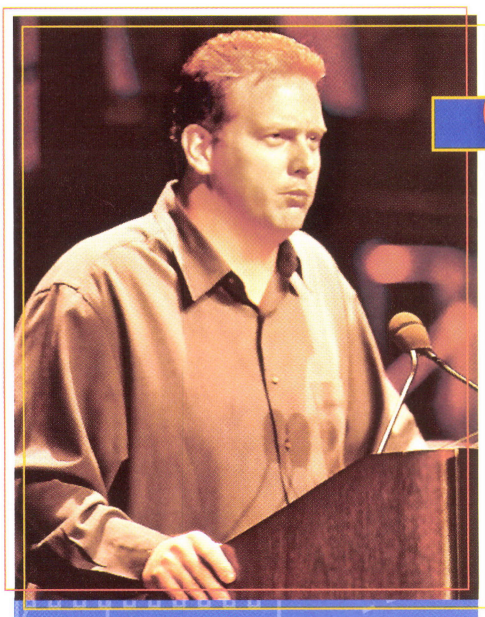
GAME INNOVATION SPOTLIGHTS
ANIMAL CROSSING by Nintendo
BATTLEFIELD 1942 by Digital Illusions
MEDAL OF HONOR: ALLIED ASSAULT by 2015, Inc.
THE THING by Computer Artworks

IGDA AWARD FOR COMMUNITY CONTRIBUTION
Doug Church

THE FIRST PENGUIN AWARD
David Crane, Larry Kaplan, Jim Levy, Alan Miller and Bob Whitehead, founders, Activision

LIFETIME ACHIEVEMENT AWARD
Gunpei Yokoi, creator, GameBoy

game developers choice awards



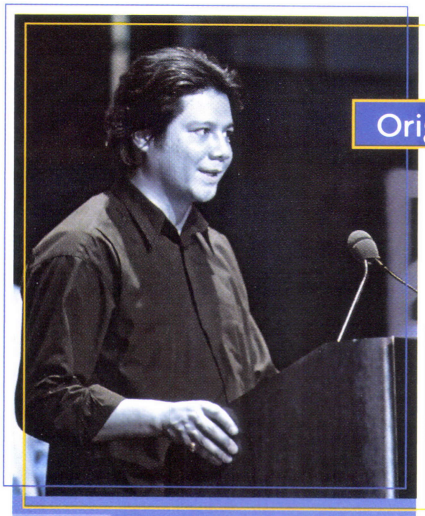
Game of the Year

METROID PRIME by Retro Studios

Development for Metroid Prime began in June of 1999 and the Retro Studios team had a daunting task ahead of them. Take one of game history's most treasured titles and convert it not only to an entirely new genre, but from a flat 2 dimensional world to a rich 3D environment. Many skeptics thought it impossible to execute with any expertise, but we haven't heard much from them since the game released – perhaps busy playing?

Based on the gameplay and structure of the original – platform action mixed with a level of combat – the mood and intense character identity bring Metroid to its next incarnation. The Retro Studios team, with some assistance from Nintendo, over deliver in every respect. Developers enjoy playing this game and selected it as Game of the Year for its high production values, well timed and engaging pace, polished graphics and gameplay and a first person viewpoint that simply draws you in. This is a reinvented classic that could eclipse it successor in sheer fan appreciation.

Producer Michael Mann accepted the award on behalf of the Metroid Prime team from Retro Studios.



Original game Character of the Year

Sly Cooper from SLY COOPER AND THE THIEVIUS RACCOONUS

World, meet Sly Cooper. Developed by the team at Sucker Punch Productions, Sly Cooper is kind of an anti-hero's hero. A charming mix of devilish and debonair, Sly is a notorious international thief with a strong sense of justice. Borrowing from a long lineage of thievery, he is out to avenge his father's death and recover the family treasure. Oh yeah, and he is a cute furry raccoon.

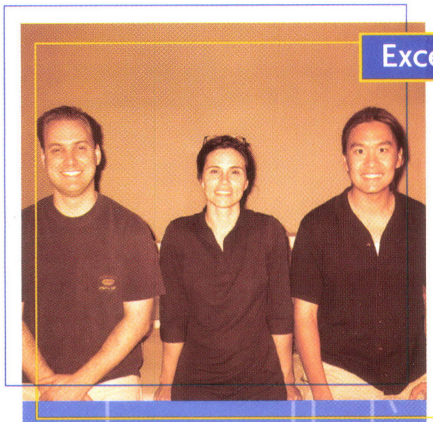
With influences from classic comic heroes and harkening back to the golden era of adorable fuzzy game mascots this character is truly unique. His complex character and attitude make him not only fun to play, but an endearing and enduring game character.

Dev Madan, Art Director for Sucker Punch Productions accepted the award for Original Character.

Rookie Studio of the Year

Retro Studios for METROID PRIME

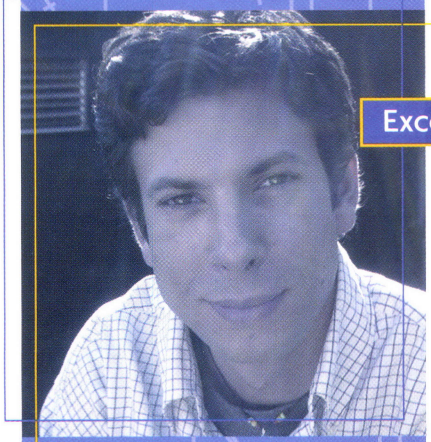
No one would accuse this year's recipients of Rookie Studio of being freshmen. A well oiled machine of veteran developers based in Austin, Retro Studios is recognized for its first offering as a collective – the beloved Metroid Prime. The studio's first commercially released title had high expectations and could have been fraught with disappointment. Instead the team, with the aid of some experienced game developers from publisher Nintendo (Shigeru Miyamoto, among others) managed not only to satisfy nostalgia, but recreate a classic into a blockbuster first person game. Such a seamless first title leaves game fans and developers alike wanting more . . .



Excellence in Audio

Jack Grillo, Rebecca Hanck, Erik Kraber and Yuan Liu for sound effects in MEDAL OF HONOR: ALLIED ASSAULT

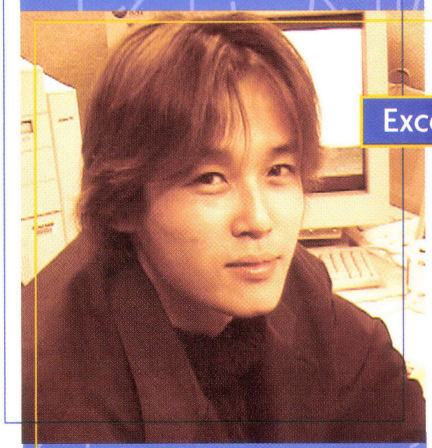
The award winning team from Electronic Arts LA knows sound – and they've got the awards to prove it. Not just recognized by the Choice Awards, the oft awarded team manages to engage all the senses and to immerse us in the world of Omaha beach and WWII completely. From the opening mission, the sound effects are that magic ingredient that makes the battle all too real.



Excellence in Game Design

Romain de Waubert de Genlis and Team for game design in BATTELFIELD 1942

Romain de Waubert de Genlis and the team from Digital Illusions brings the second world war to your computer and you to historical battles all over the globe. The carefully crafted game design lets players choose characters, national loyalties, sidearm weapons and battle locations. Multiplayer teamwork is an integral part of a successful campaign. The game designers expertise combine a brand new game engine capable of dynamic models and landscapes, ground, sea and air physics, and an unparallel variety of game play to fully immerse the player in the greatest battle of the last world war.



Excellence in Visual Arts

Tetsuya Nomura for art direction in KINGDOM HEARTS

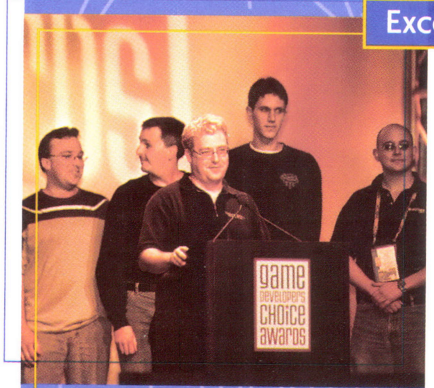
"Unlock your Destiny" is just the marketing tag line for Kingdom Hearts, but one that Art Director Tetsuya Nomura took seriously. He managed to seamlessly blend some of animation's most classic and beloved characters with what is the recognized standard for Japanese RPG. Talk about a challenge: there are no two greater powerhouses than Disney and Squaresoft, in their respective entertainment industries. The fans for both border on cult. Yet Nomura's art direction brought these two worlds together with beautiful and entertaining results.

Excellence in Level Design

Metroid Team for level design in METROID PRIME

The level of detail required to transition from a 2D platform to an all immersive 3D world is immense. But that's all part of reinventing a classic. The Metroid team creates level after level, world after world that remain authentic to the Metroid purist and engages the first person player. Thanks to the level design team no detail is left to chance.

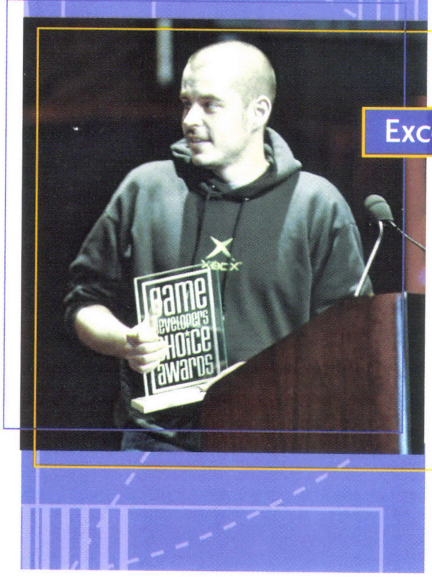
Producer Michael Mann accepted the award on behalf of the Metroid Prime team from Retro Studios.



Excellence in Programming

Mark Brockington, Scott Greig, Jason Knipe, Don Moar and Don Yakielashek for network programming in NEVERWINTER NIGHTS

Multiplayer has changed gaming forever, and in some wonderful instances has caused us to redefine games we've long loved. If NeverWinter Nights is a game that brings the huge medieval fantasy world of Dungeons and Dragons to the computer, the BioWare programming team brings the players to one another. These avid players are networked together in a flawless and engaging manner. The programmers manage to blend the best of single-player and massively multiplayer games, as well as the classic experience of pen-and-paper role-playing D&D, to create the best of all possible worlds.

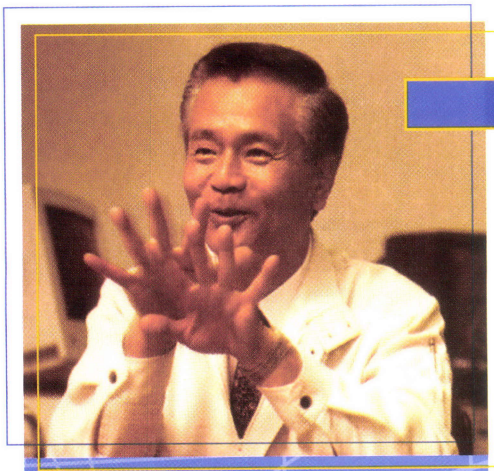


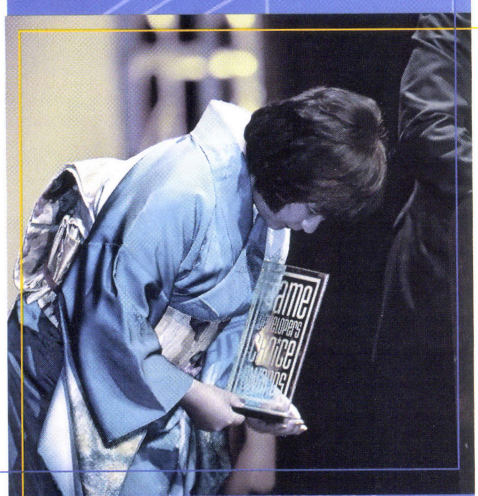
Excellence in Writing

Clint Hocking and JT Petty for writing in TOM CLANCY'S SPLINTER CELL

This is the first Excellence in Writing Award received and it seems somewhat poetic that it go to a developer who created a game insired by a renowned author. For games, good dynamic writing has been an elusive challenge. Clint Hocking and JT Petty illustrate that that is a thing of the past. Players caught up in Splinter Cell enjoy a sophisticated story — much in the Clancy style — screenplay like dialog and a natural shifting and twisting plot.

Clint Hocking accepted the Choice Award for Excellence in Writing.





Lifetime Achievement Award

Gunpei Yokoi

The Lifetime Achievement Award was presented by former recipient Yuji Naka. To best represent the evening and honor we've reproduced, from the original Japanese, Mr. Naka's tribute.

"Tonight we honor one of the great pioneers in our industry, Gunpei Yokoi.

Yokoi-san is a founder and visionary of our industry. Without his contributions, games could not have become the important part of our culture that they are today. We are all indebted to him for making our passion popular and viable.

As the head of Nintendo's research and development team he first created a small credit card sized game platform called Game and Watch. This was just his first success to make our electronic games mobile and adopted by a mass audience.

Yuji Naka

As mentor to the young Nintendo prodigy Shigeru Miyamoto, Yokoi-san and Miyamoto formed a creative partnership that resulted in some of the classic arcade games of the 1980's: Donkey Kong, Donkey Kong Jr., Mario Brothers and the first Metroid.

In 1989 Yokoi-san and his team developed the most popular and successful hand-held game console ever – to date the GameBoy platform and its successors have sold over 142 million units worldwide.

Presenting this award tonight is very meaningful for me. Gunpei Yokoi has been a great inspiration throughout the 20 years that I have spent creating games. Although SEGA and Nintendo were fierce competitors in the past, the works of Gunpei Yokoi continually motivated and challenged me to do my best work and to strive to achieve greatness with all my games. Gunpei Yokoi is a man who has not only inspired the imagination of game developers around the world, but he has changed the way the world plays.

Tragically, Yokoi-san's brilliant career was abbreviated when he passed away 1997. His legacy remains for all of us to build upon. To accept his honor for Lifetime Achievement are his wife and family from Kyoto, Japan."

The son and widow of Gunpei Yokoi accepted his honor to a standing ovation.

The Lifetime Achievement Award is selected by the Choice Awards Advisory Board.

The First Penguin Award

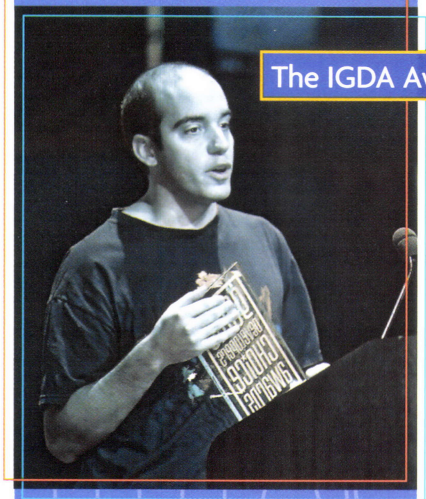
David Crane, Larry Kaplan, Jim Levy, Alan Miller and Bob Whitehead, founders, Activision

The recipients of the First Penguin Award are the founding team of Activision who established the first 3rd party developer. In 1979 the Atari ex-patriots created a model that celebrated, recognized and rewarded the game designers directly and continues as an industry standard today.

Among their game credits Dragster, Crane's adaptation of the 1977 Atari/Kee coin-op Drag Race, as the first game independently released for the Atari VCS, in 1980, followed closely by Checkers, Boxing and Fishing Derby. 52 games in total were released by the company between 1980 and 1988, with the designers' identities prominently featured in all packaging and advertising.

David Crane, Larry Kaplan, and Bob Whitehead accepted the Choice Award for the founding team.

The First Penguin Award is selected by the Choice Awards Advisory Board and recognizes the courage and bravery of a developer who tested the proverbial "waters", uncertain of success or failure. A "first penguin" serves as a lesson, and inspiration, to the rest of the community.



The IGDA Award for Community Contribution

Doug Church

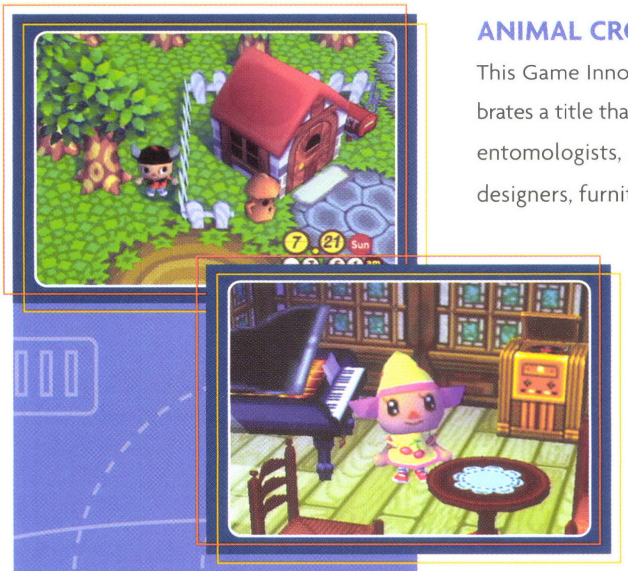
The IGDA honors Doug Church for his many and varied contributions to the game development community and the art form of games. Named one of GameSpy's 30 Most Influential People in Gaming, Doug's contributions to the community also include serving on the GDC Advisory Board for the last 3 years. He encourages and betters others with his numerous articles and GDC talks on improving design methods. As co-chair of the IGDA's Education Committee, Doug is a continual advocate for industry/academia relations.

Not limited to community support only, Doug's career inspires others by his innovative work. Game credits include; Frequency, Deus Ex, System Shock, System Shock 2, Thief: The Dark Project, Thief II, Flight Unlimited, Flight Unlimited II, Terra Nova: Strike Force Centauri, Ultima Underworld, and Ultima Underworld 2.

The IGDA Award for Community Contribution is selected by the Choice Awards Advisory Board.

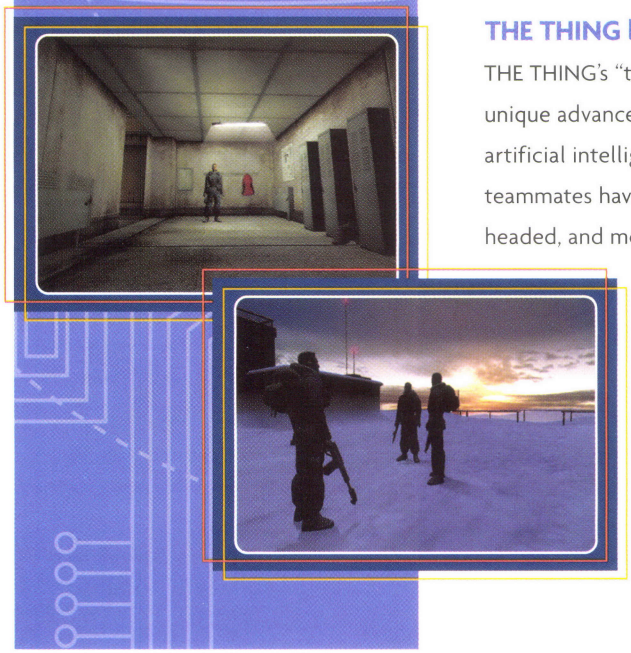
Game Innovation Spotlights

Game Innovation Spotlights honor games with that *certain something*. These are the games developers themselves talk about, admire, anticipate — play. Innovation keeps our art form vital and evolving. Each of the four games honored with a Choice Award for Game Innovation, in its own unique way, changes the way we make games.



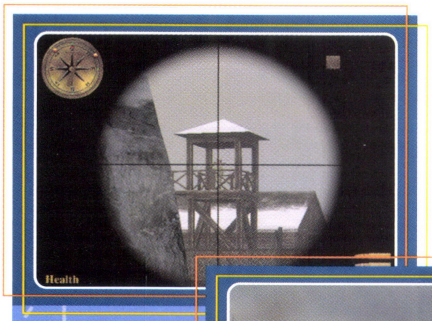
ANIMAL CROSSING by Nintendo

This Game Innovation Spotlight award celebrates a title that appeals to aspiring fishermen, entomologists, archaeologists, fashion designers, furniture collectors, essayists, personal assistants, and gardeners. The sandbox-style gameplay unfolds in time with clock and calendar. Memory cards enable a robust community and trading economy to flourish entirely offline. Integration with the GameBoy Advance and E-Reader offers players a constant supply of new experiences. ANIMAL CROSSING's innovative community-based gameplay is as easy to pick up as it is difficult to categorize.



THE THING by Computer Artworks

THE THING's "trust" and "fear" systems are unique advancements and a true indication of the advances in artificial intelligence. To date most NPC (non-player character) teammates have been intermittently helpful, frequently dunder-headed, and mostly virtual scenery. How you influence NPCs' psychological state in THE THING determines whether or not these characters will cooperate. For the first time they express fear, question your trust, and keep a close virtual eye on your actions, incorporating a whole new dimension of play.



MEDAL OF HONOR: ALLIED ASSAULT
by 2015, Inc.

MEDAL OF HONOR: ALLIED ASSAULT is honored for its immersive gameplay. This Game Innovation honoree has an introduction like no other. WWII is a rich environment for realism, the opening mission at Omaha Beach is one of the most frightening and intense game experiences ever created. With just one mission, this game demonstrates our medium's unique ability to engage the player through gameplay alone.

BATTLEFIELD 1942
by Digital Illusions

This Game Innovation Spotlight Award recognizes BATTLEFIELD 1942 for advancement in cooperative multiplayer gaming and variety in gameplay styles. Set in the battlefields of WWII, cooperative multiplayer isn't just fun, it's a necessary component to success. You can parachute, drive tanks, fly airplanes, or engage in hand to hand combat. This unprecedented variety of gameplay styles converge with solid engine technology to create a truly innovative first person shooter.

"I started in this industry about 15 years ago with the idea to do something different, something that hasn't been done before. That still drives me and the rest of the team . . . to be recognized for doing something that we thought was fun is truly an honor."

— Fredrik Liliegren of Digital Illusions, Choice Award Recipient for Game Innovation

Three Years of Honors

GAME OF THE YEAR

2001 THE SIMS by Maxis
2002 GRAND THEFT AUTO III by DMA Design/Rockstar Games
2003 METROID PRIME by Retro Studios

ORIGINAL GAME CHARACTER OF THE YEAR

2001 Seaman from SEAMAN
2002 Daxter from JAK & DAXTER: THE PRECURSOR LEGACY
2003 Sly Cooper from SLY COOPER AND THE THIEVIUS RACCOONUS

ROOKIE STUDIO AWARD

2001 Counter-Strike team for COUNTER-STRIKE
2002 Bohemia Interactive Studio for OPERATION FLASHPOINT
2003 Retro Studios for METROID PRIME

EXCELLENCE IN AUDIO

2001 Matt Uelmen, Jason Hayes, Glenn Stafford & Andrea Pessino for composition in DIABLO II
2002 Marty O'Donnell & team for sound effects in HALO: COMBAT EVOLVED
2003 Jack Grillo, Rebecca Hanck, Erik Kraber and Yuan Liu for sound effects in MEDAL OF HONOR: ALLIED ASSAULT

EXCELLENCE IN GAME DESIGN

2001 Harvey Smith & Warren Spector – for game design in DEUS EX
2002 GTA3 team for game play in GRAND THEFT AUTO III
2003 Romain de Waubert de Genlis and Team for game design in BATTLEFIELD 1942

EXCELLENCE IN LEVEL DESIGN

2001 American McGee, Jim Molinets & team for level design in AMERICAN MCGEE'S ALICE
2002 Fumito Ueda & team for level design in ICO
2003 Metroid Team for level design in METROID PRIME

EXCELLENCE IN PROGRAMMING

2001 Sims programming team for artificial intelligence in THE SIMS
2002 Richard Evans for artificial intelligence in BLACK & WHITE
2003 Mark Brockington, Scott Greig, Jason Knipe, Don Moar and Don Yakielashek for network programming in NEVERWINTER NIGHTS

EXCELLENCE IN VISUAL ARTS

2001 Ryuta Ueda & Kazuki Hosokawa for art direction in JET GRIND RADIO
2002 Fumito Ueda & team for art direction in ICO
2003 Tetsuya Nomura for art direction in KINGDOM HEARTS

EXCELLENCE IN WRITING

2003 Clint Hocking and JT Petty for writing in TOM CLANCY'S SPLINTER CELL

LIFETIME ACHIEVEMENT AWARD

2001 Will Wright
2002 Yuji Naka
2003 Gunpei Yokoi

THE FIRST PENGUIN AWARD

2001 Chip Morningstar & Randy Farmer
2002 Hubert Chardot
2003 David Crane, Larry Kaplan, Jim Levy, Alan Miller and Bob Whitehead

IGDA AWARD FOR COMMUNITY CONTRIBUTION

2001 John Carmack
2002 Jeff Lander
2003 Doug Church

GAME INNOVATION SPOTLIGHTS

2001 COUNTER-STRIKE by Counter-Strike team
CRAZY TAXI by Sega AM3
DEUS EX by ION Storm Austin
JET GRIND RADIO by Smilebit
NO ONE LIVES FOREVER by Monolith
2002 BLACK & WHITE by Lionhead Studios
GRAND THEFT AUTO III by DMA Design
ICO by Sony Computer Entertainment, Inc.
MAJESTIC by Electronic Arts
REZ by United Game Artists
2003 ANIMAL CROSSING by Nintendo
BATTLEFIELD 1942 by Digital Illusions
MEDAL OF HONOR: ALLIED ASSAULT by 2015, Inc.
THE THING by Computer Artworks

"I'm really really proud to be a part of this development and pushing toward a great new way to tell stories, which I think is an important part of being a human being."

— Clint Hocking, Choice Awards Recipient for Excellence in Writing in TOM CLANCY'S SPLINTER CELL

9 years of GameDeveloper for just $229

Each CD-ROM contains:

- ✓ Full text of feature articles, columns, and product reviews
- ✓ Articles, artwork, and diagrams exactly as they were printed
- ✓ All code and applications that accompanied the articles

Great content. Great prices.

NEW!	✓ May 2002 to April 2003 (Available June–preorder today.)	*$49*
	✓ May 2001 to April 2002	*$39*
	✓ May 2000 to April 2001	*$39*
	✓ May 1999 to April 2000	*$39*
	✓ 1994 to April 1999	*$99*

Or get everything:

✓ Premier issue to April 2003	*$229*

Order by phone:
800.444.4881 (US) or 785.841.1631 (international)
Order online:
www.gamasutra.com/store

Sponsored by: Maya™ Alias | wavefront

GamaNetwork

the game. When a playable prototype is developed, any technological or strategic concepts can be tested for viability. But the benefits go far beyond that.

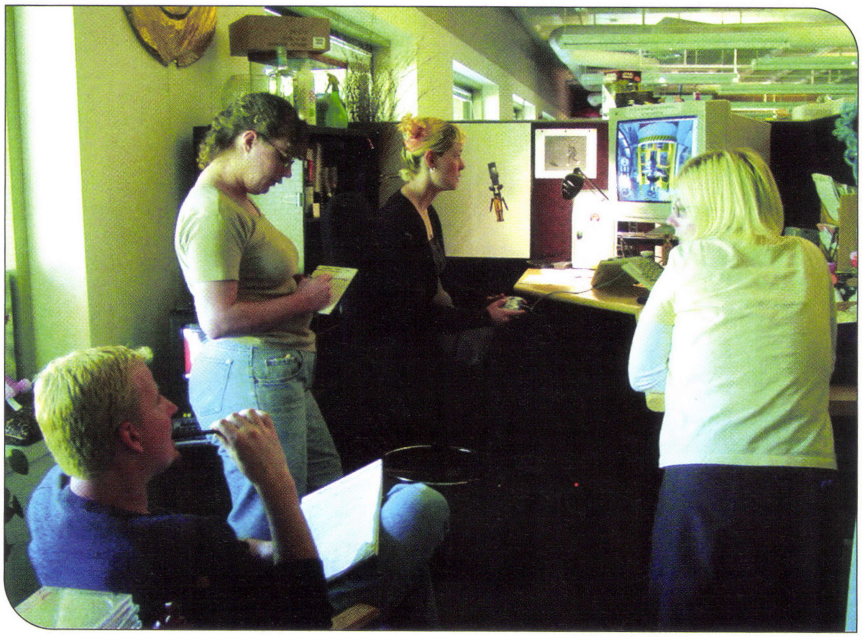
With a prototype, all critical game elements can be analyzed before they are invested in too greatly. These elements are all created during preproduction with a smaller team, which should save the cost of developing them. Any major obstacles should be found early so that fixes or alternate paths can be determined before the production is fully underway. Since the preproduction team is small, it also gives the core team time to gel together and figure out their processes before taking on the added responsibilities of team management.

Play-testing. Once a playable build exists for a game, play-testing can begin. During a play-test, someone plays a game and offers feedback on their experience with it. This can be done with in-house participants or out-of-house participants and can range from formal to informal.

For instance, at LucasArts there is an informal in-house play-test group. Anytime a team wants feedback on their product, they can call on a group of volunteers to play their game for them. Recently, play-testers were requested to play an original action-adventure title, RTX Red Rock. Volunteers were scheduled for sessions at a level designer's desk and asked to play a specific level of the game for two hours while members of the game team watched and took notes. The participants were asked to think out loud while they played so that the team could gain better insight into what choices they were making as they progressed through the assigned level.

The game team felt that the feedback improved the levels greatly, and plan to do more testing in the future. Overall, the cost and time involved was low, and the impact on the game was high. Mainly, the team laments not having begun these tests earlier and not having enough time to address everything the tests revealed about the game.

Play-testing can help teams in seeing how actual players interact with the product. The team is too close to the design to see objectively, and the data collected from these exercises can be very enlightening.



Lily Childs thinks aloud while playing RTX Red Rock during an in-house LucasArts test, as team members Harley Baldwin, Tim Miller, and Shara Miller take notes.

Offering the opportunity for open-ended feedback at the end of a play-test can elicit even more insight into the experience with the product. Asking questions such as "What did you like best (or least) about the game?" can allow teams to anticipate what is or is not fun about the game at a point when gameplay can still be adjusted.

Advanced Measures

In-house play-testing provides valuable feedback, but the information gathered still isn't as accurate or helpful as it could be if greater resources were available. This type of testing is not truly reflective of a target audience, because even if the participants fall into the correct demographic (genre of game, age, gender, and so on), they are still coming to the test with much more game knowledge than an average consumer would have. They know more about games and game development in general, and they are within the company, which means they inevitably are bringing some preconceived notions about the product with them to the play-testing session.

Play-testing with outside users. Using outside participants instead of people in-house requires greater resources. First of all, compensation is required to encourage people to attend sessions. Money, copies of games, or other company memorabilia can serve this purpose. The second expense is the overall effort it takes to schedule participants in a timely way.

Before testing begins, the facility in which the participant is to be tested has to be determined. They can simply play at a designer's desk, although such a setup potentially skews the data. Participants may feel compelled to try to please the designer, since he or she will be present, which can alter how they evaluate the product and what sentiments they choose to express while thinking aloud. The designer may also find it extremely difficult not to intervene in the test in some way. To keep designers from altering the test process and results, they ideally

Microsoft Game Studios User-Testing Group Play-test Lab cubicle.

should observe from a separate room either through one-way glass or on video.

Play-testing doesn't have to be a one-on-one experience with a designer. A participant can play a portion of the game for a given time frame and then offer feedback through a questionnaire. Once participants get a taste of the game, they can offer a subjective perception about the product's overall fun factor.

Questionnaires. Creating a scientifically sound questionnaire is a tricky business and ideally requires a psychology expert. Depending on how a question is phrased, it will generate a different answer. In order to get the most accurate data possible, questions have to be developed with the least likelihood of leading the person to answer in an anticipated or predictable way. Getting someone to create questionnaires and analyze the data is expensive. The data resulting from this type of test, though, will be quantitative and better able to guide the design revision process. Knowing the percentage of people who felt positively or negatively toward certain facets of the game can help the team prioritize which changes to make in the amount of time available.

Structured usability evaluations. While play-testers can work en masse in a big enough lab, structured usability tests are run individually. In a structured usability test, information can be acquired about efficiency and effectiveness, rather than satisfaction as in play-tests. If designers want to know whether users are learning required tasks as they are playing, whether tasks and or levels can be completed without too much confusion, or whether a control scheme feels intuitive, structured usability testing is a great method to use.

To use this method, tasks to be performed are established prior to testing. Usually, someone familiar with the game will run through the tasks and establish an ideal time for each one. Then, a user is asked to perform them while thinking aloud. As the participants progress, the following is noted: whether they can successfully accomplish each task, how long they take to accomplish it, and anything they say or do to indicate how they are working to achieve each goal. In the usability field, six to eight users can uncover the vast majority of usability problems with each task. While it's promising that so few users are needed to find the majority of design problems, there is no guarantee that the solution implemented for any problem is successful unless it too is tested.

For game development, a variation called RITE (Rapid Iterative Testing and Evaluation) might solve some of the time issues with finding and resolving usability issues using the standard method. In this technique, developed by Microsoft Game Studios User-Testing Group (see For More Information), a participant engages in particular tasks and then the design is changed immediately based on the results before another user is run. This way, a set of users are not experiencing the same problems over and over again, but rather verifying whether past issues have been resolved. This method yields fast results and makes the most of each participant's involvement, but requires the development team to have time as testing occurs to make the ongoing changes.

Expert evaluations. Beyond user-testing

FOR MORE INFORMATION

GAME-SPECIFIC USABILITY RESOURCES

Microsoft Game Studios User-Testing Group:
www.microsoft.com/play-test/publication

Federoff, Melissa. "Heuristics and Usability Guidelines for the Creation and Evaluation of Fun in Video Games." Master's thesis, Indiana University, 2002.
www.melissafederoff.com/thesis.html

GENERAL USABILITY RESOURCES

Dumas, J., and J. C. Redish. *A Practical Guide to Usability Testing.* Norwood, N.J.: Ablex, 1993.

Nielsen, J., and R. Mack. *Usability Inspection Methods.* New York: John Wiley and Sons, 1994.

Norman, Donald. *The Design of Everyday Things.* New York: Doubleday, 1990.

Jakob Nielsen's Web Site
http://useit.com

ONLINE GUIDE TO USABILITY RESOURCES

www.usabilityfirst.com

methods there are expert evaluation methods, which can be utilized by usability professionals. These evaluations do not require users, but do require someone who knows how to anticipate user behavior. While these methods do not yield data with the same amount of validity as user testing, they can help to uncover usability problems in a faster, cheaper way when necessary.

One of the most common expert evaluation methods used by usability professionals is the heuristic evaluation. Heuristics are agreed-upon standards that are used to evaluate a design. To do a heuristic evaluation, the usability professional makes one pass through the product to become familiar with it, and then makes a second pass to determine whether it is meeting or failing each heuristic. Jakob Nielsen has created heuristics for software interfaces, but an agreed-upon list of heuristics does not exist yet for games. Table 1 shows one possible list of heuristics for games.

Another expert usability evaluation method that could be applied to games is the cognitive walkthrough. During this technique the usability professional walks through a scenario and tells a convincing story about whether the determined path for players would be the one they would actually take. This would be a particularly strong method to use with games that have a linear structure to make sure that the expected path for the players is the one they will likely follow.

Raising the Bar

The field of usability offers developers many methods to determine the efficiency, effectiveness, and satisfaction of product designs throughout development. The data resulting from the empirical testing of users can help designers to make informed decisions and improve the overall quality of their games. Many games are already taking advantage of usability testing to improve player experience and satisfaction, which will ultimately raise the bar for all games in terms of consumers' expectations.

GAME HEURISTICS

1	Game Interface	Controls should be customizable and default to industry-standard settings
2	Game Interface	Controls should be intuitive and mapped in a natural way
3	Game Interface	Minimize control options
4	Game Interface	The interface should be as non-intrusive as possible
5	Game Interface	For PC games, consider hiding the main computer interface during gameplay
6	Game Interface	A player should always be able to identify their score/status in the game
7	Game Interface	Follow the trends set by the gaming community to shorten the learning curve
8	Game Interface	Interfaces should be consistent in control, color, typography, and dialog design
9	Game Interface	Minimize the menu layers of an interface
10	Game Interface	Use sound to provide meaningful feedback
11	Game Interface	Do not expect the user to read a manual
12	Game Interface	Provide means for error prevention and recovery through the use of warning messages
13	Game Interface	Players should be able to save games in different states
14	Game Interface	Art should speak to its function
15	Game Mechanics	Mechanics should feel natural and have correct weight and momentum
16	Game Mechanics	Feedback should be given immediately to display user control
17	Game Mechanics	Get the player involved quickly and easily
18	Gameplay	A clear, overriding goal of the game should be presented early
19	Gameplay	There should be variable difficulty and multiple goals for each level
20	Gameplay	"A good game should be easy to learn and hard to master" (Nolan Bushnell)
21	Gameplay	The game should have an unexpected outcome
22	Gameplay	Artificial intelligence should be reasonable yet unpredictable
23	Gameplay	Gameplay should be balanced so that there is no definite way to win
24	Gameplay	Play should be fair
25	Gameplay	The game should give hints, but not too many
26	Gameplay	The game should give rewards
27	Gameplay	Pace the game to apply pressure to, but not frustrate the player
28	Gameplay	Provide an interesting and absorbing tutorial
29	Gameplay	Allow players to build content
30	Gameplay	Make the game replayable
31	Gameplay	Create a great storyline
32	Gameplay	There must not be any single optimal winning strategy
33	Gameplay	Use visual and audio effects to arouse interest
34	Gameplay	Include a lot of interactive props for the player to interact with
35	Gameplay	Teach skills early that you expect the players to use later
36	Gameplay	Design for multiple paths through the game
37	Gameplay	Players should be rewarded with the acquisition of skill
38	Gameplay	Build as though the world is going on whether your character is there or not
39	Gameplay	If the game cannot be mode-less, it should feel mode-less to the player

TABLE 1. Game heuristics developed during a thesis case study performed by the author. They were compiled after reviewing relevant literature and observing and interviewing a development team. They are a starting point for discussion; further research is required in order to verify them. The project is available online in its entirety at www.melissafederoff.com/thesis.html.

Switch to Discreet 3ds max™ and tower above the rest.

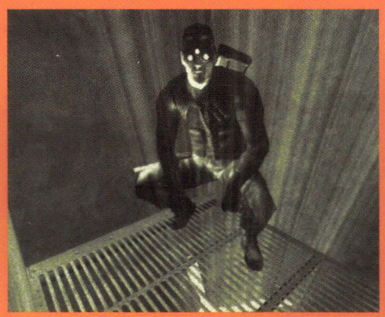
With tight deadlines and multi-platform development challenges, creating a hot-selling game can drive anyone over the edge. Make great games even better by leveraging **3ds max** and its tight game engine connectivity, award-winning character and game play design tools like UVW Unwrap, Render-to-Texture and Radiosity, and the flexibility to fit into any pipeline.

Whether it's PlayStation®, Xbox™, GameCube™ or PC development, step onto safe ground with **3ds max** and the world's largest collection of scripts, plug-ins, partnerships and user communities.

The world's top games depend on **3ds max**, how about you?

For more information or a 30-trial of **3ds max** go to www.discreet.com/products, or for a demonstration call 1.800.869.3504.

Panzer Dragoon Orta courtesy of SEGA and Smilebit/SEGA



Tom Clancy's Splinter Cell™ courtesy of Ubi Soft Entertainment

Battlefield 1942 courtesy of Digital Illusions

3ds max™ 5
discreet®

Images courtesy from left to right Panzer Dragoon Orta Original Game © SEGA CORPORATION © Smilebit Corporation/SEGA CORPORATION, 2002, Tom Clancy's Splinter Cell™ - © 2002 Ubi Soft Entertainment and Battlefield 1942™ - © 2002 Digital Illusions. All rights reserved.

Discreet is a division of Autodesk, Inc. Discreet, and 3ds max are either registered trademarks or trademarks of Autodesk, Inc. in the U.S.A. and/or other countries. All other brand names, product names or trademarks belong to their respective holders. © 2003.

Beautiful, Yet Friendly Part 1:

Stop Hitting the BOTTLENECK

A couple of years ago I was driving home to Quebec when I stopped near the Ontario border to gas up. I got out of my car to stretch and noticed two other travelers engaged in a complicated mish-mash of hand waving and broken English. I approached, thinking I could help the poor fellows by acting as a translator between both parties, when I realized that not only were they both French Canadians but neither of them knew it.

If I found the situation amusing at the time, I've since come to realize that specialized jargons aren't so different from languages as different as French and English. Like languages, jargon plays a key role in communicating information about a given discipline; but like languages, jargon can also erect artificial barriers between initiates and neophytes.

Game development studios comprise numerous different professions, each with their respective jargons. In the time I've spent bridging the gap between art and code at Pseudo Interactive, it has often been challenging — but ultimately much more fulfilling — to try and explain concepts to artists.

As any experienced game artist knows, dealing with a disgruntled graphics programmer mumbling about "vertex diets" isn't the most enthralling part of the day. In our programmer minds, every model should be hand-tuned and lovingly optimized, and every artist should know the intricacies of the hardware and software they are dealing with.

But in the end, a programmer's job isn't to make a good-looking game, it's to empower the artists to make a good-looking game. In this two-part article on graphics performance, I will attempt to explain how to manage performance without losing track of this vision statement. In this month's part, we will take a look at the big picture and explain how you can often solve performance problems without optimizing a single mesh. We will also see when, if, and what to optimize in your scene. In next month's conclusion, we will get down to the details and study how certain modeling practices can improve the performance of your meshes with minimal impact on their visual quality.

It's Not Art's Fault, It's Level Design's!

I was happily debugging the other day when an artist dropped by my desk and asked me how many vertices he could pack in a cubic meter without encountering performance problems. Put on the spot, I simply replied, "Ask level design." Although I intended that as a joke, it wasn't exactly bad advice. Programmers can accurately predict

GUILLAUME PROVOST | *Originally hired at the age of 17 as a lowly system programmer writing BIOS kernels for banks, Guillaume has been trying to redeem himself ever since. He now works as a 3D graphics programmer at Pseudo Interactive and sits on his local Toronto IGDA advisory board. You can contact him at depth@keops.com.*

how many vertices you can pack into view, but how many "cubic meters" are actually in view at one time is largely determined by level design, not code.

Viewed in this light, level designers operate in a kind of performance macrocosm, and their work provides your best indication of the performance constraints in which you are operating. In fact, on a performance scale, it's a lot easier for things to go wrong in level design than in art. Highly occluded environments that rely on portals and potential visible sets, for example, can suffer greatly from having several portals aligned into view. And instancing a mesh a thousand times in a room isn't likely to be performance friendly no matter how optimized it is.

Balancing level complexity and prop complexity is generally the combined job of the game designers and level designers, but as an artist you'll benefit greatly from knowing in which contexts objects are used. Carefully optimizing meshes can be a tedious, thankless task. Making sure you pay attention to those objects that count most first will get you the most out of the system for the least amount of work.

At Pseudo Interactive, levels are always hashed out first using rudimentary geometry. Once the level has been play-tested and the rough placement of game-logic items has been ironed out, it gets shipped to the art team for beautification. By the time artists start populating a room with art assets, they already know they will have to minimize decorative details if there are, say, 10 highly tessellated bad guys in a room.

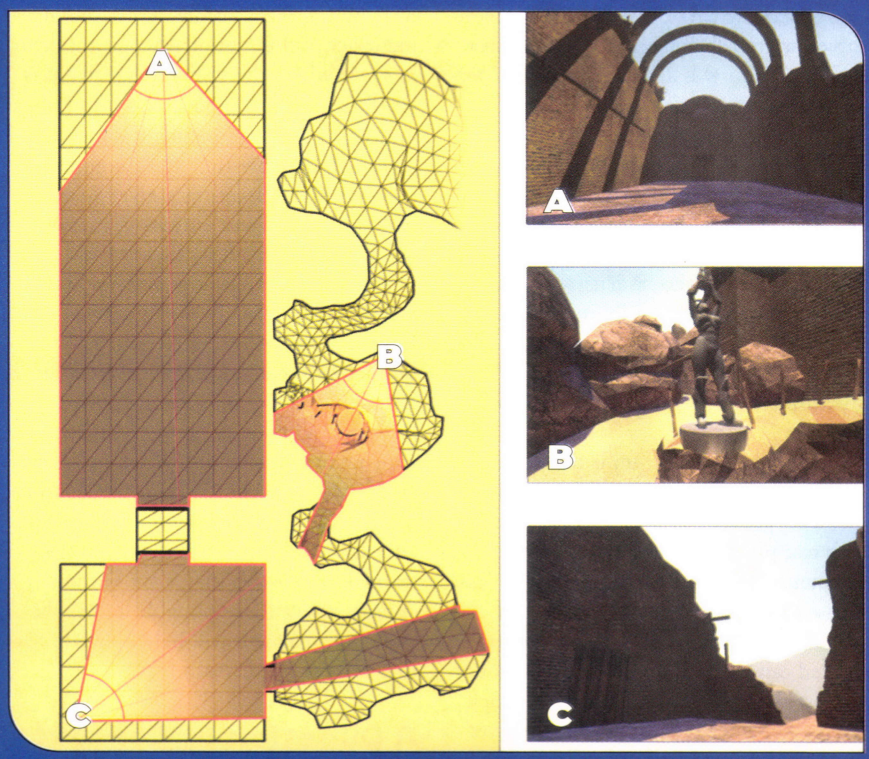


FIGURE 1. The visibility spectrum in a portal-enabled visibility engine. In camera A and B, the visibility spectrum is closed off using a door that disables the portal. In camera C, transition zones keep the player from seeing into the next area. The very small visibility spectrum in C lets us place a highly detailed statue that we could not afford in the other areas.

Murphy's Law

Performance is like laundry: it's a chore, and people only notice it if it's a problem. If your clothes are clean 95 percent of the time, people will only remember — and judge you by — those 20 days out of the year where your clothes were not clean.

Frame rate is a direct function of all objects in view at a given time, and as such it is most likely to go down when the visual and environmental stimuli are at their peak. Since, in most games, that also happens to be both when players are most enjoying themselves and when they require the most responsiveness out of the system, performance hits can be significant sources of player frustration.

Provided that you've been somewhat reasonable in building the art assets of a scene, getting good performance out of it is more about avoiding or offsetting the impact of worst-case scenarios than it is about just "making things go faster." No matter what environment you're building art for — a heavily occluded starship or a snowy medieval landscape — always assume players will position themselves in the worst vantage point possible in terms of performance. If there's a single spot in the entire level the player can sit at and bring all its glorious complexity into view at one time, you can rest assured that players will strive to get to it.

Great Wonders Get Separated

There are two increasingly common schools of thought in how the art-development workflow works. In the first, art builds a series of set pieces, and then level design assembles and populates levels using them; this method is not unlike a Lego puzzle. In the second scenario, level design builds basic proxy geometry and populates proxy levels with all logic-related entities, then hands it off to art for embellishment.

But whether you are a level designer constructing a level from existing build-

ing blocks, or part of an art team responsible for embellishing a proxy level, the only way you will achieve a constant performance benchmark is to balance your scene complexity correctly. We will see later that there are several factors that may or may not influence scene complexity, but for the purpose of a high-level assessment, you should take into account three things: the vertex density, the texture density, and the visibility spectrum.

*Scene Complexity ≈ Vertex Density * Texture Density * Visibility Spectrum*

The visibility spectrum is the set of all visible space from a given location. Since the total vertex count and texture space you can put into the visibility spectrum is constant, the larger that space is, the less dense the detail you can pack in it. If you're authoring art for a mostly unoccluded outdoor environment, the amount of vertex and texture data you can pack per cubic meter is probably much lower than it could be if you're authoring contents for heavily occluded interior environments (assuming your engine has some form hierarchical visible surface determination system for occluded environments). In fact, the visibility spectrum is the single most important aspect affecting rendering performance.

The smaller you make your visibility spectrum, the more detail you can put in your scenes, and the better your frame rate will be. Typical examples of techniques commonly used to decrease the size of the visibility spectrum include closing off rooms with doors or using transition zones that block off the view for indoor environments (Figures 1a and 1b), and using fog or depth-of-field effects in external or other typically unoccluded environments.

Vertex density is how tightly packed vertices are in a given volume of space. If small areas of the world contain many highly detailed objects, your performance is likely to drop off when they come into view. If you have a few art assets that are particularly expensive, distributing them evenly across the playing space and intentionally placing them in areas where the visibility spectrum is small (Figure 1b) is both faster and more important than optimizing the individual pieces. Remember that scene complexity depends on all assets in a given area, including intelligent entities that may have roamed into it by themselves.

Texture density is how much texture memory you actually use in a given area. Like vertices, textures can be a severe bottleneck when you concentrate a lot of different textures into a constrained location. When such areas come into view unannounced, all the texture data concentrated in the said area needs to be downloaded into the graphics processor, and this can cause breakdowns in the frame rate. Distributing your texture density as evenly as possible across the visibility spectrum will help alleviate texture-related bottlenecks.

A Question of Context

So you've examined your level and objects at a high level and fixed what could be fixed, but for gameplay, design, or artistic reasons, you are stuck with certain situations where you need to go down to the object level and start making things go faster.

Michael Abrash once said that it's more important to know what to optimize than how to optimize. The rule unequivocally rings true in programming circles, but it applies to graphics-related content just as well as it does to graphics code. If you're running into performance problems, blindly optimizing meshes without any previous evaluation of which ones require it is akin to putting all your clean dishes in with the dirty ones each time you run your dishwasher.

In fact, unless you are not expected to go anywhere near your performance limits — a common, if dubious, assumption — there is a subset of your art assets that you should always carefully optimize. If you're working on a third-person game, chances are your main character will get a lot of attention, a lot of detail, and undergo a lot of iterations. Since the character is always in view, its visual quality bar is much higher. But if always being on-screen is a good reason to lovingly fine-tune a mesh, it's also a prime reason to ensure it is performance friendly. Objects that are always or frequently in view are generally the most detailed, most expensive entities in the game, and, by the very fact of their presence on-screen, they also constantly eat up processor time. Objects that are instanced a lot may also make their way onto the screen more often than is readily apparent. Always make sure that you carefully balance visual quality and performance when authoring such content.

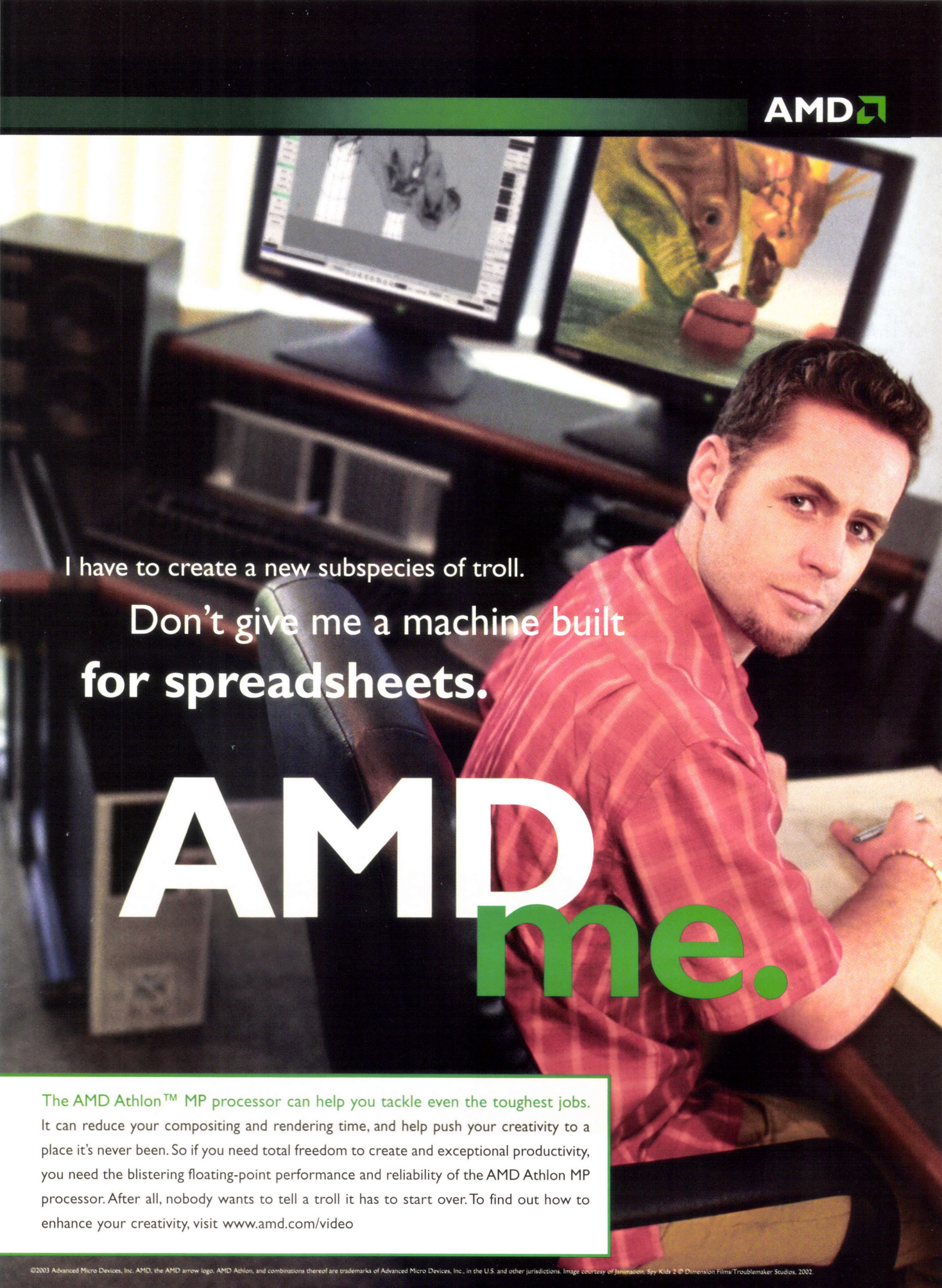
Since frame rate typically goes down in spurts, common sense dictates that you should be wary of objects that are placed or used in situations that already stress the capacities of the system. Good examples of these include opponents and combat effects. Objects of this nature tend to turn performance problems into performance nightmares, so make sure you identify all stress situations and make their related art assets as performance friendly as possible.

Finally, if your performance problem arises in a specific area, then you will want to hit the most expensive objects in that area first. But how does one properly estimate the cost of an object?

Transform-Bound, Fill-Bound

The introduction of programmable vertex and pixel pipelines and the new ability of hardware to blend several texture layers in a single pass has made estimating the rendering cost of an object increasingly difficult. I often found myself giving contradictory guidelines to our art team and on several occasions could not formulate clear explanations on what factors came into play and why, until I realized that, in the end, it came down to a fairly simple set of rules.

The first thing to remember is that


AMD
I have to create a new subspecies of troll.
Don't give me a machine built
for spreadsheets.
AMD me.
The AMD Athlon™ MP processor can help you tackle even the toughest jobs. It can reduce your compositing and rendering time, and help push your creativity to a place it's never been. So if you need total freedom to create and exceptional productivity, you need the blistering floating-point performance and reliability of the AMD Athlon MP processor. After all, nobody wants to tell a troll it has to start over. To find out how to enhance your creativity, visit www.amd.com/video
©2003 Advanced Micro Devices, Inc. AMD, the AMD arrow logo, AMD Athlon, and combinations thereof are trademarks of Advanced Micro Devices, Inc., in the U.S. and other jurisdictions. Image courtesy of Jamination. Spy Kids 2 © Dimension Films/Troublemaker Studios, 2002.

graphics processors do not understand or care about objects. They deal with portions of an object on a material-by-material basis. If you want to think in terms of performance, the first thing you should do is to decompose your object into its individual materials, then assess the resulting parts one by one. Programmers call the resulting parts surfaces.

There are two fundamental things that happen when rendering a surface: its vertices get transformed, and its triangles get drawn. At any one time, both operations happen in parallel, so that whichever of the two is slowest (drawing the triangles or transforming vertices) is roughly how fast a mesh can be rendered (Figure 2).

When assessing a surface's optimization needs, you need to identify both its bottleneck and its cost. The cost will tell you whether you need to optimize it, and the bottleneck will tell you what to optimize. If an object's bottleneck is transform time, then it is said to be "transform-bound," and its cost is the time it takes to transform the vertices attached to it. If an object's bottleneck is fill time, then it is said to be "fill-bound," and its cost is the time it takes to draw the surface on-screen.

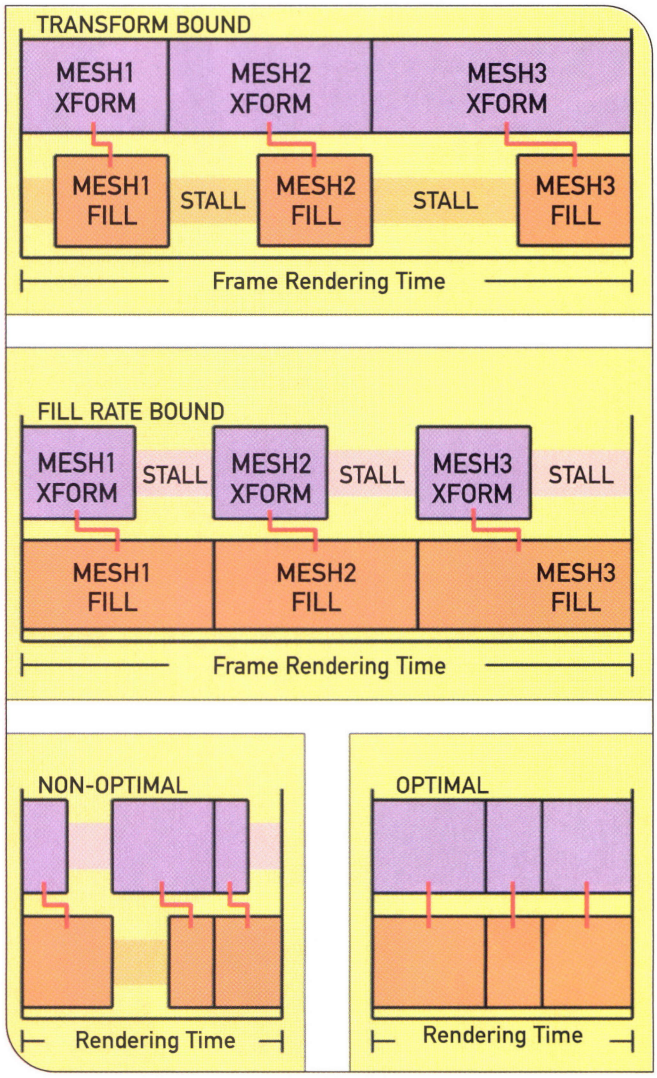


FIGURE 2. Transform/fill parallelism.

Game artists therefore need to develop an intuitive sense of whether a surface is likely to be fill-bound or transform-bound. Fill-bound meshes have different (and sometimes opposite) optimization rules from transform-bound meshes. So in order to model — or optimize — efficiently, one must decide on which set of rules to follow by assessing whether transform or fill is likely to be the bottleneck.

In practice, interactive articulated bodies are generally transform-bound, walls and ceilings tend to be fill-bound, and props lie somewhere in between. But as we'll see, other factors need to be considered when choosing which optimization route to take.

To Screen Space We Shall Travel

Computational transform cost, the time it takes to transform the vertices, depends on two factors: the transform complexity and the number of vertices to transform. Since programmers like to have nice formulas, I'll attempt to give an abstraction of the transform cost here:

*Transform Cost ≈ Vertex Count * Transform Complexity*

The transform complexity is how long it takes to convert individual vertices into screen space. It's highly dependent on what kind of transforms your rendering system supports. In Pseudo Interactive's first title, CEL DAMAGE, we supported (in order of increasing complexity) static objects, binary-weighted bones, seven-point quadratic FFD (free-form deformation) cages, and morph targets.

As a simple example of this, a static wall or ceiling has a very low transform complexity, while a fully articulated zombie with morphing goose bumps has a very high transform complexity. If your renderer can have multiple lights affect an object, then the complexity of the lighting conditions (how many lights, what type of lights) also affects your transform complexity.

The second factor in the equation is vertex count. Surfaces that have a lot of vertices also have a higher total transform cost. If you are authoring content using higher-order surfaces, such as Bézier and B-spline patches, or if you are using hardware-accelerated displacement maps, then your total vertex tally will grow as a function of the tessellation level individual primitives undergo. (All three methods also entail high transform complexity costs.)

Otherwise, surfaces that have very low vertex counts are unlikely to have a high transform cost, unless the transformations they undergo are exceptionally

www.microsoft.com/corpevents/meltdown2003

meltdown 2003

Two intensely hot days of tracks for Game Programmers, QA/Compatibility Technicians, Technical Artists, Business Developers and Publishers

Microsoft® Windows®

Gaming & Graphics

July 16 & 17

Seattle, WA

The Westin Seattle

July 29 & 30

London, UK

The Royal Lancaster

Microsoft®

© 2003 Microsoft Corporation. All rights reserved. Microsoft, Windows, and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

complicated or numerous. Weight-blended morph targets, a very high number of bones, or higher-order FFD cages such as tri-cubic (64 points per cage) and tri-quadratic (27 points per cage) solids would fall into that category. Surfaces that have both high vertex counts and a high transform complexity can easily produce major bottlenecks.

Now, I've talked about transform costs, but not about the likelihood of a surface being transform-bound. A fork lying on a table might contain few vertices, but the cost associated with drawing such a tiny object is so small, that transform might still be the bottleneck. Hence:

Transform-Bound Likelihood ≈ Transform Cost / Fill Cost

If the fill cost of a mesh is smaller than the transform cost, an object will be transform-bound (it will take longer to transform) no matter how small the transform cost is (or no matter how large the fill cost is).

The bottleneck associated with transform-bound surfaces is related to vertices and transformations, so their performance is generally not affected by the nature of their material or the size of their textures (however, on consoles that require vertices to be re-sent every frame, texture uploads can compete with vertex uploads for bandwidth). Performance can, however, be affected by the number of successive texture stages applied to the surface: for each platform, there is a hard limit on the number of textures it can blend together before it needs to make a second rendering pass. A separate rendering pass entails retransforming all the vertices of the surface, effectively doubling the transform cost associated with the mesh.

Mainstream consoles and PC video cards support anywhere between one and eight textures per pass. Since the number of texture passes a material requires is highly specific, you should consult your programmer to find out when you are causing extra passes to occur.

Fat Triangles Make for Fat Fill Times

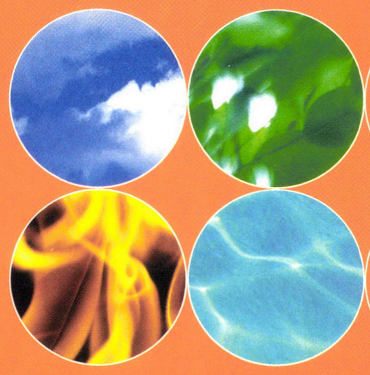
So what about your average ceiling or wall? Fill time (or draw time) depends largely on the size of the surface on-screen, the number and size of textures involved, and the draw complexity:

*Fill Cost ≈ Pixel Coverage * Draw Complexity * Texture Density*

The draw complexity is the complexity of the operations that occur every time a pixel gets drawn. It is typically a function of how many texture passes are involved and what kind of mathematical calculations occur with respect to those passes. In general, sophisticated per-pixel lighting effects, such as bump and normal maps or spherical harmonics, tend to have high draw complexities that grow increasingly complex with the number of lights by which they are affected. Materials that distort the view, such as refractive glass, or materials that cast volumetric shadows, also have very high fill costs. Since you can often combine several such properties on a single surface, the fill cost of multi-pass surfaces can go through the roof.

Even if your draw complexity is pretty tame, it's easy to forget the potential bottleneck caused by texture size and density. A wall filled with a giant mural, a high-resolution light map, and a detail map to dirty it up might screw with your texture cache because of the high volume of texture memory to which it refers.

No matter what, your surfaces are generally unlikely to have high fill costs if they are small on-screen. But since triangle sizes are pixel-based, higher resolutions or FSAA (full-screen anti-aliasing) modes will directly affect your fill rate by enlarging every triangle's pixel area. The



MAKE BETTER GAMES

The game industry is comfortably in the middle of a console lifecycle, PC graphics technologies continue to advance at a rapid rate, and new opportunities including mobile gaming continue to offer developers an array of options.

Yet significant challenges face developers today. How do small development houses survive in a high stakes environment? How do large shops streamline their production methods and pipelines efficiently and optimally? Which technologies and processes should be integrated to yield the best results?

Moving forward, the time to prepare for the next generation systems is now, before business plans are written, and well before development kits arrive. How will the next generation affect game development and what unforeseen issues will arise?

GDC Europe addresses the serious issues facing game developers today and tomorrow. Four days of thought provoking keynotes, lectures, tutorials, panels and roundtables provide relevant and essential information and inspiration. Join Europe's leading developers to explore the future of development and make better games.

GameDevelopers Conference Europe

26-29 August, 2003 • London, UK

www.gdc-europe.com

In partnership with

ects 2003
28-30 august earls court london uk

Part of

LONDON GAMES WEEK
25-31 AUGUST 2003

higher the resolution, the more important fill rate becomes. In fact, enabling FSAA is a very common technique to estimate roughly how much of a scene is fill-bound.

Surfaces taking up a lot of screen space with either high draw complexities or a lot of texture data can be major bottlenecks. Although you can't exactly make your walls and ceilings any smaller than they are, you should be wary of the material properties you set on those surfaces.

The fill-bound likelihood is the inverse of the transform-bound likelihood:

Fill-Bound Likelihood ≈ Fill Cost / Transform Cost

If the transform cost of a mesh is smaller than the fill cost, an object will be fill-bound (it will take longer to draw) no matter how small the fill cost is (or no matter how large the transform cost is). In any case, if both transform cost and fill cost are low, you should skip the object and concentrate on something more problematic.

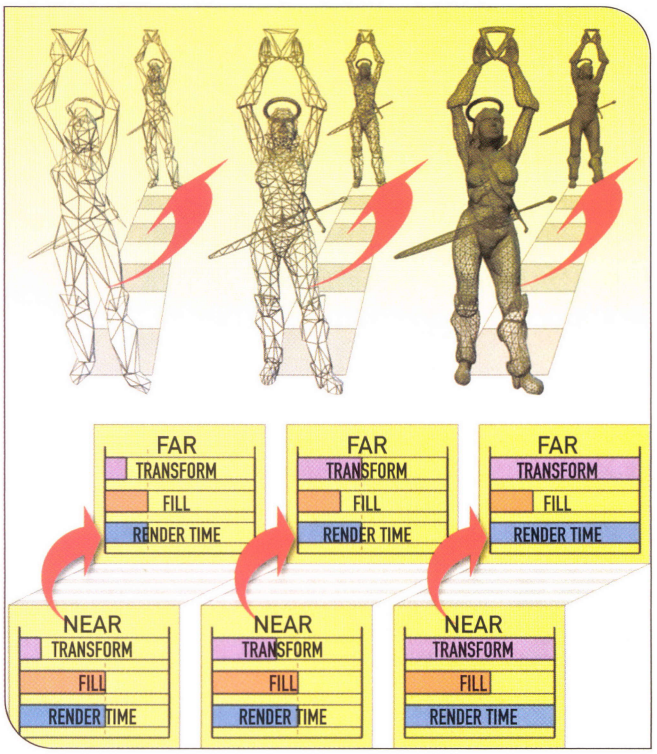


FIGURE 3. The fill cost of a mesh decreases with distance, while the transform cost does not. This can cause meshes to shift from being fill-bound to being transform-bound as they recede into the distance (center). If a mesh has a very high vertex density, then its cost may stay the same regardless of distance (right). To reduce the associated transform cost of a mesh in conjunction with its diminishing fill costs, lower-level-of-detail meshes are used (left).

You may have noticed that the expansion of the fill-bound and transform-bound likelihood equations promptly throws us into programming geek-world. We can abstract it further by extricating the two most significant components out of the equation: vertex counts and pixel coverage. This gives us vertex density, which is really just the vertex count divided by the screen size of an object.

Transform-Bound Likelihood ≈ Vertex Density

Although this is an extremely simplified abstraction of the fill-bound, transform-bound question, it makes approximating the answer somewhat more manageable: screen vertex density is your best — if not sole — indicator in distinguishing transform-bound surfaces from fill-bound surfaces.

From Fat to Flat and Big to Small

As an artist, it can be difficult to predict the on-screen vertex density of an object during play: as objects get farther into the distance, the on-screen vertex density rises. This can cause large variations in vertex and texture densities to occur when objects are viewed at different distances from the camera (Figure 3).

There are two very important techniques that exist to equilibrate an object's vertex and texture density through distance metrics: discrete levels of detail, and texture mip-maps. Discrete levels of detail (or LOD meshes) replace the full-detail versions of a mesh when it is sufficiently far away from the viewer that the switch produces negligible differences in the rendered image quality. Similarly, mip-maps are lower-detail versions of a texture that can be used when the texel-to-pixel density is sufficiently high. Well-constructed mip-maps will actually enhance your image quality and are an absolute prerequisite to making fill-bound surfaces performance friendly.

Although both mip-maps and LOD meshes tend to have a greater impact in outdoor environments, where the very large visibility spectrums call for techniques to minimize the impact of high-detail objects, they both are important to learn and use in all situations.

Skimming vertices and textures out of your mesh will never hurt. But, as we'll see next month, optimizing a mesh for best performance is not a piece of cake. So if you're going to go to lengths to make certain objects truly performance friendly — and in certain cases, you should — then you should build at least one proper level of detail for them first.

Finally a surface can also cycle between being transform-bound and fill-bound if its vertex density is very nonuniform (Figure 4). Surfaces with both very high and very low curvature areas or surfaces that are adaptively tessellated for lighting conditions typically suffer from this problem. When dealing with such surfaces, remember that it's important to save vertices in the high-density areas, not in the low-density areas where the renderer is likely to be fill-bound. Better yet, try to distribute your vertex density as equitably as possible across its surface area, balancing the load between transform and fill, and maximizing your use of both processing pipelines.

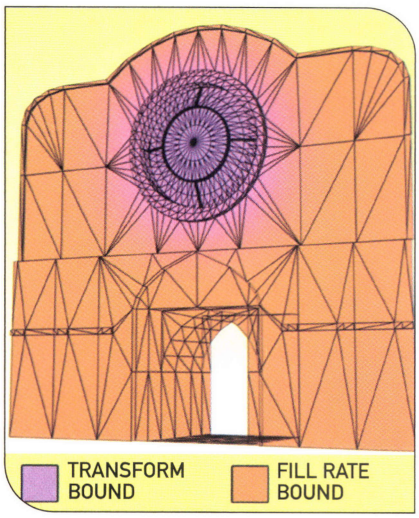


FIGURE 4. Non-uniform density mesh.

When Big Is Too Big

There's an exception to this whole system: objects that are always only partly visible.

If you were to merge all rooms and corridors of a level into a single object, its resulting vertex density might be low, but you would only ever draw a very small portion of it at every frame. When a surface comes into view, only the visible portion needs to be drawn, but all its vertices need to be transformed and all its textures need to be sent to the card.

If — and only if — you have objects that have either a significant amount of vertices or large amounts of texture data, and if those objects are too big ever to be entirely in view at once, then you should break them up into smaller pieces. If you do break an object up because of its size, try splitting it in roughly equally distributed volumes to maximize culling efficiency. If your object has a great deal of different materials, try to break up the object in a way that balances the texture load across chunks.

Last week, our art staff built a skybox. The texture they applied on it was — quite understandably — very detailed. Since its refined color gradations did not palletize very well, we were suddenly stuck with a 512K texture gobbling up almost all of our available texture memory every frame. We correspondingly split the skybox into four sections, so that at any one time, only half of the original texture was required on-screen, unless you looked straight up. Since there isn't all that much to render when you look straight up, splitting in this case was the right call.

Microscopes and Binoculars

You'll avoid many performance headaches by paying attention to context and scale. If certain props are consistently located far away from the in-game camera, then you should naturally give them less detail.

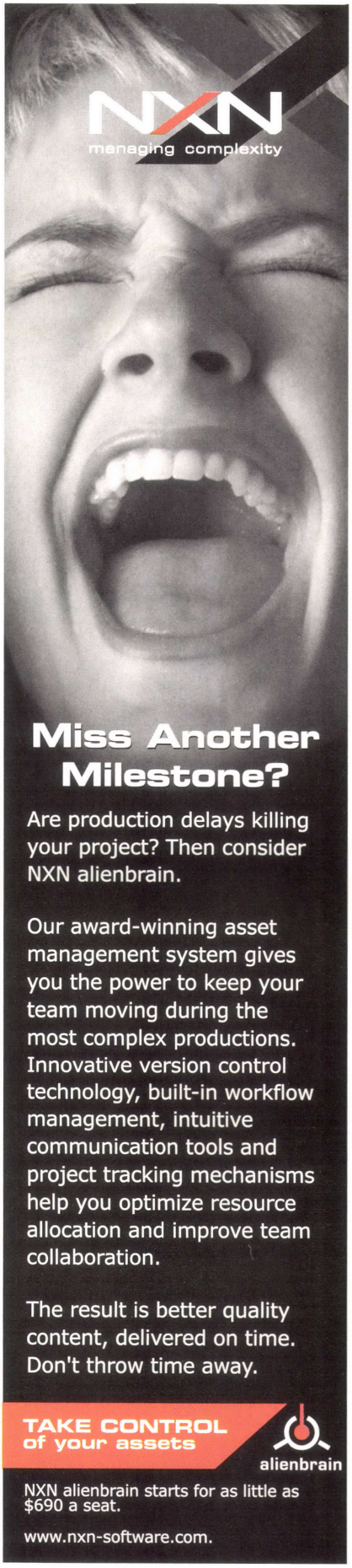
Similarly, chances are that small objects will be small on-screen too. If you model objects without a scale reference, you are more likely to spend your vertex budget on a scale where the detail will be lost on the player.

Next Month

You've analyzed your scene, made reasonable distribution decisions, and identified certain objects as needing an optimization pass. You found what bottlenecks and costs were likely to be — but then what? And how does this all affect day-to-day work? In next month's conclusion, we will explore the hands-on modeling and texturing techniques that can be used to reduce cost for both transform-bound and fill-bound surfaces.

ACKNOWLEDGEMENTS

Thanks to Danny Oros for providing both the illustrations and several rounds of feedback. Thanks also to Benoit Miller from Matrox, Sébastien Dominé and Sim Dietrich from Nvidia, Guennadi Riguer from ATI, and Ryan McLean from MagiTech, for kindly answering questions and reviewing the article. And last but not least, thanks to the rest of the Pseudo team, for playing the guinea pigs on this article.



Insomniac Games' Ratchet & Clank

GAME DATA

PUBLISHER: Sony Computer Entertainment America
NUMBER OF FULL-TIME DEVELOPERS: 40
NUMBER OF CONTRACTORS: 1
LENGTH OF DEVELOPMENT: 18 months
RELEASE DATE: November 5, 2002
TARGET PLATFORM: Playstation 2
DEVELOPMENT HARDWARE: PS2 Dev Tools, PCs: avg. dual 800MHz–1.2GHz with 1GB RAM
DEVELOPMENT SOFTWARE USED: Insomniac's own tools suite, Maya, Photoshop, ProDG, MS Visual Studio, CodeWright, Deep Paint, ProTools, Sound Forge, Premiere, Illustrator
NOTABLE TECHNOLOGIES: Much of the engine technology was developed in-house, but some very important renderers were developed by Naughty Dog; sound licensed from Sony

The scene: Twenty developers lounging on a sun-drenched porch overlooking Barham Boulevard in Los Angeles, drinks in hand, enjoying the warm breeze and listening to traffic rumble by below. The occasion: Our first post-SPYRO brainstorming meeting.

It was late spring 2000, and even though we were still in production for SPYRO: YEAR OF THE DRAGON (our last SPYRO), we knew we had to start planning for our first PS2 project. Our problem was twofold: we had decided not to develop any more SPYRO games, and we were deciding whether we wanted to stay with the platform-action genre. It's a familiar scenario for game developers: the road is wide open, but figuring out which direction to travel is excruciating.

We had meeting after meeting trying to narrow down the choices — and with 20 people involved, things got tense and sometimes depressing. I was driving hard to move us away from the platform genre because Al Hastings, our vice president of technology, had very astutely suggested that this was the perfect opportunity not only to expand our abilities but to address other niches in the console market currently overlooked by U.S. developers.

After coming up with and discarding countless ideas, we settled on a concept best described as a dark adventure. We wanted to try a game with a bit more realism and immersion than our previous efforts. This meant moving away from bright environments, cartoony characters, and platform mechanics. This also meant creating a macro design and story that were far deeper than those of the SPYRO series.

We called the concept "I5" (for Insomniac game #5), and the main character was a human girl with a staff. She would fight with the staff as well as use it to activate magic with special katas — martial arts moves performed using directional input. There was a strong Mayan influence to the overall look of the game, and the characters and environments we planned were more realistic than anything we had attempted since our first game, 1996's DISRUPTOR.

We pitched our game idea to SCEA and were fortunate to strike a deal very early in preproduction. Once we had Sony's backing, our preproduction team dove in and began working on PS2 technology, final macro design, and all of the elements that would help us create our first playable.

Within a couple of months, however, it was clear that things weren't going well.

First, we couldn't nail down the main character. She was too cartoony, and then too mundane; the colors we chose ended up looking weird on-screen, and we couldn't get the proportions right. In the past, proportion had never been a problem, since we had always worked

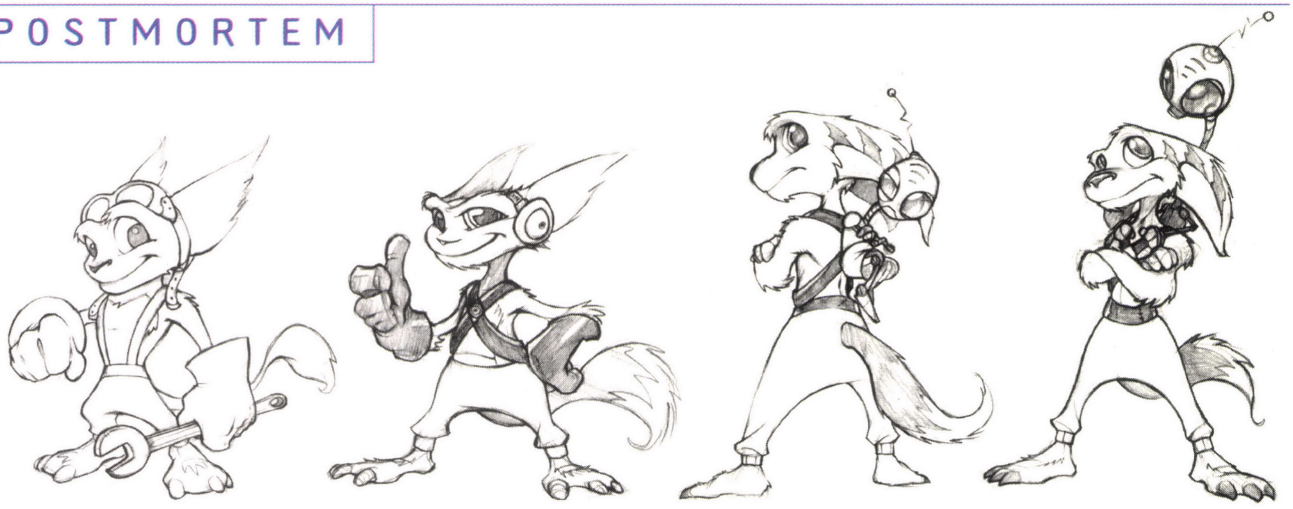
TED PRICE | *Ted is president and founder of Insomniac Games.*

with nonhuman characters. But we quickly realized that it's easier to spot flaws in human characters than in nonhuman ones. Even though our main character eventually looked acceptable, she still lacked that *je ne sais quoi* which would make her stand out.

Then there was the hardware. We were making the jump from PSX to PS2 in very little time, and Al Hastings was shouldering the entire burden with some help from Mark Cerny, who had written the original VU code used on the first-ever PS2 engine. Al and T.J. Bordelon, tools programmer, were, at the time, trying desperately to get the engine and tools to the point where the artists could use them to build and prototype environments and characters. Looking back, I can't believe they actually got everything to work, and work well, in a matter of months. Still, the technology was not yet state-of-the-art, and we all wondered how it would fare against the second generation of PS2 titles.

But the worst part of the process was the entire team's ambivalence about the project. No one was truly excited about the game or where it was heading. We were making it work through sheer effort. My job was to be the concept's champion, but maintaining a positive demeanor was proving more and more difficult. Morale was at its lowest in Insomniac's nine-year history.

We eventually ground out a first playable, and while it wasn't bad, it



Some early concept sketches for Ratchet.

wasn't great either. And we wanted something great. Our Sony producers, who were very polite about their reservations, confirmed our feelings. Nonetheless, they had reservations. At one point Connie Booth, our SCEA executive producer, suggested that we might want to rethink the direction we were taking. While being very clear that Sony would support us with whatever we decided, she pointed out that not only would the PS2 adventure category be crowded upon our planned release date, she also believed that we were no longer playing to our team's strengths.

After digesting her words, Al Hastings, Brian Hastings — Insomniac's vice president of programming — and I (the three partners in the company) did some soul searching and realized that Connie was right. By pushing on, we could release a solid adventure game, one that might even do well. But slogging through another year of developing a game no one was excited about would kill the team.

So on March 20, 2001, we stopped preproduction of I5 and started over. We would be going back to our forte, action-platforming. This announcement moved the team's mood lever from reverse to overdrive. Everyone was energized and excited about the new prospects.

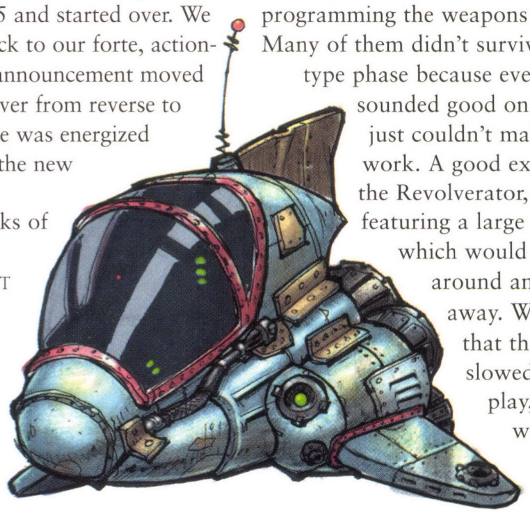
Within two weeks of this decision, we developed RATCHET & CLANK's basic concept. In a matter of days, Dave Guertin, our lead character designer, nailed the two main characters, and soon we were brainstorming on the weapons and gadgets that players would be using.

Once we got started, we never looked back. That isn't to say problems didn't exist during the process, but it was the best and most enjoyable production experience we've had at Insomniac.

What Went Right

1. Prototyping. We had been prototyping gameplay since SPYRO THE DRAGON, but never to the extent that we did with RATCHET & CLANK. The game featured more than 35 weapons and gadgets, all of which had to be fun to use. The big problem we faced was that every weapon and gadget was woven into the macro design and the story. If we had to pull one out during production, the macro design would collapse, which would be disastrous for the production schedule.

We spent three months building and programming the weapons and gadgets. Many of them didn't survive the prototype phase because even though they sounded good on paper, we just couldn't make them work. A good example was the Revolverator, a weapon featuring a large drill bit which would spin enemies around and fling them away. We discovered that the spinning slowed down gameplay, and that it was difficult to hit enemies, since the collision for the drill bit had to be narrow to be believable. Another good idea on paper was the Mackerel 1000, a fish that would be a replacement for Ratchet's wrench. It sounded funny, but when we put it in the game the humor lasted for about three seconds.



We also prototyped enemy layouts and behavior to a much greater extent on this project. The majority of our enemies were well tested and tuned before each level went into production. This process saved us a massive amount of time, since we only built final models and did final coding once we were sure that the enemies would work. Conversely, on the Spyro series we were always ripping things out and starting over during production, since we rarely prototyped gameplay. With RATCHET & CLANK, and for all of our future projects, gameplay prototyping has now become an ongoing process.

Finally, to clearly establish the look of the game, we used our I5 engine to prototype two of the game's planned environments before we had the real RATCHET & CLANK technology up and running. It was all smoke and mirrors, but it allowed us to show on-screen what we imagined the final game would look like and put to rest a lot of our own fears about whether or not the game would stand out visually.

2. Sharing technology with Naughty Dog. Shortly after we decided to start over, Jason Rubin, Naughty Dog's co-founder, called me and asked if we'd be interested in checking out the technology they developed for JAK & DAXTER. He explained that Naughty Dog didn't want anything from us other than a gentlemen's agreement to

Photographer: Daniel Gebhardt - Source: BiblePlaces.com - Copyright © 2002 Todd Bolen

Build your game with care and Thot™.

For game developers,
by game developers.

Unified
- asset management
- version control
- process control

Customizable
- client
- server
- workflow

Affordably priced.

thot-tool.com

Thot™ built by Redshift Software, Inc.

share with them any improvements we made to whatever we borrowed plus any of our own technology we felt like sharing. In an industry as competitive as ours, things like this just don't happen.

We went over to Naughty Dog's offices and took a look, particularly at their background renderer. They had developed some incredible proprietary techniques to render smoothly transitioning levels of detail and instanced objects very quickly. We brought the code back to our offices, spent some time getting a handle on their techniques, and then we were up and running with a much more powerful environment engine.

Needless to say, Naughty Dog's generosity gave us a huge leg up and allowed us to draw the enormous vistas in the game. In return, we've shared with them any technology in which they were interested, but so far we've been the clear beneficiary of the arrangement.

3. Setting reasonable design goals. Even though the concept behind RATCHET & CLANK was ambitious for us (integrating RPG elements into an action-platformer), we were careful not to cram too much stuff into the initial design.

We had never made a game before where we didn't have to axe one or more levels at some point in the production process because we were out of time. The RATCHET & CLANK macro design was more complex, so we couldn't afford to rip out a level at the last moment. Sony had created a tremendous marketing campaign that relied on a specific release date, so missing our delivery dates was not an option. Plus, we were already releasing pretty late in the year, and to miss one week of precious pre-Christmas sales would prove very costly.

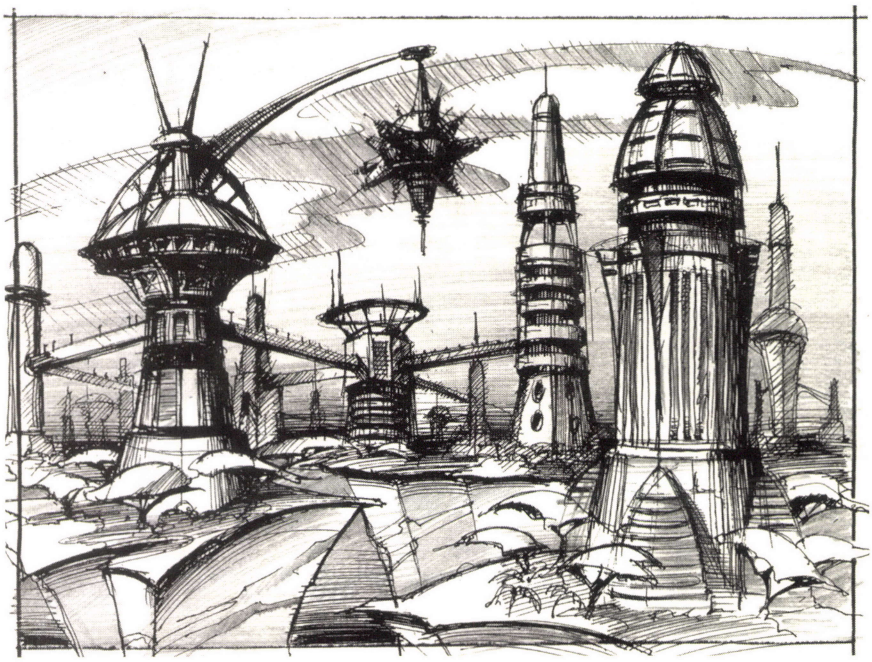
For these reasons, we planned the game layout much more carefully than we had on past titles. We had a pretty good idea of how long it would take to build each level, but we also knew that plenty would go wrong during the production process. So even though we had time to do 20 levels, we cut back to 18 at the very beginning.

We also made sure that nothing went into the design unless we were very sure that it was going to work. Early prototyping was key here, but so was an attitude of general restraint. There were a few wild concepts that everyone was excited about, but had we integrated them into the macro, the project probably would have slipped. Ultimately we were able to put about 90 percent of what we planned into the game — a record for us.

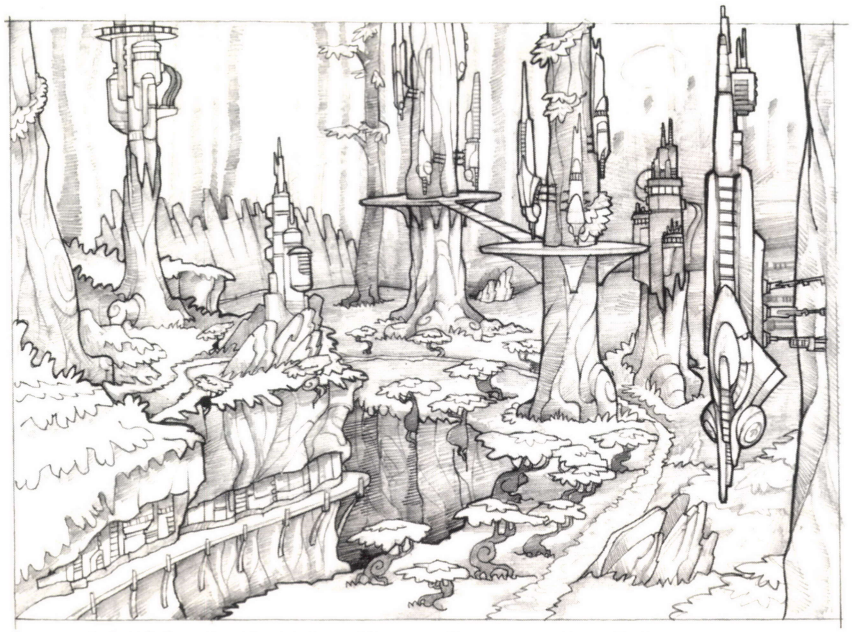
4. Focus testing. Most games go through focus testing at some point. Publishers and developers alike want to see how people react to the game and whether it's too difficult or too easy. Because it's the best way to tune the gameplay, we've focus-tested our games since the first SPYRO. But with RATCHET & CLANK we went overboard.

We had four major focus tests during production. Each focus test featured another 25 percent of the game until we were testing the full game at alpha. More than 200 consumers got to play the game before release, and the feedback we collected was invaluable. By recording and charting data from the game, we were able to tune item prices, adjust challenge difficulty, and change monetary rewards. Without this exhaustive process the game would probably have been unplayable.

Just as important, though, was the fact that each focus test forced us to get the game working. Along with the other deadlines it sometimes felt that we were



One of the game's early production design sketches.



A concept sketch for a location that would eventually serve as planet Eudora.

always in crunch mode. The gameplay programmers in particular lived a nightmare existence between fixing bugs for the next focus disc and trying to move ahead with the new levels. But the constant burns kept us on track and on schedule. Given RATCHET & CLANK's scope and complexity, if we had waited until the end of the project to burn playable discs, the bug list would have been overwhelming and we would have missed our ship date by months.

5. Collaborative design. Everyone in the company has always been free to contribute creatively to the projects. It's not a requirement, but for those who are interested it's an opportunity to affect the direction our games take. Programmers are encouraged to contribute to story, artists are asked for ideas on design, and so on. During RATCHET & CLANK, a large percentage of the team contributed ideas outside of their particular areas of expertise, making the game one of the deepest and most varied titles we've developed.

This does not imply that we design by consensus. There's a solid structure in place to ensure that we adhere to the macro design and remain consistent with the game's "flavor." But adopting an approach that encourages design participation gives us a real wealth of creativity from which to draw while enhancing the sense of ownership everyone feels in our games.

What Went Wrong

1. Poor disc-burning process. Making the switch from CD-ROMs on the PSX to DVDs on the PS2 sounded like it would be easy. After all, we survived the challenge of recording PSX discs with quirky burners and non-intuitive burning software. What we didn't account for was the incredible amount of time that building and burning the DVDs would take.

We had to first transfer the code and data to the PC on which we would generate the files necessary to create a playable disc. Next we'd have to transfer the files to the burner PC. Then the burner software would have to create a disc image, and finally we could burn the disc. By the end of the project we were working with 4GB of data. Combining those steps with slow connections and a burner that we had to use at only double speed to prevent errors, the entire process took more than four hours to generate one disc. And there were many, many places along the way where something could go wrong, forcing us to start over again.

There were countless instances where a level would be out of memory or someone would change the memory card format, breaking everything. But we wouldn't know about it until the final disc had popped out of the tray and had been booted up on a test station. Two mistakes like this would cost an entire day.

So why didn't we change the process? Based on our PSX-burning experience, where the system was extremely finicky, when we had things working on the PS2 we didn't want to touch it and risk breaking everything. This was especially true near the end of the project.

As a result, a few of us didn't go home for days at a time near the end of the project. I remember promising our testers that if our first gold burns worked, I would do circuits of the office singing Britney Spears songs as loud as I could. Fortunately for everyone in the office, they didn't.

The result of our disc-burning pain is that we've now completely overhauled our system. We believe we've halved the overall disc production time for our current project.

A panoramic shot of Kyzil Plateau, on Planet Veldin.

Rendering of the Gagdgetron Headquarters, located on Kalebo III.

2. Late start on cinematics. RATCHET & CLANK has a much more lengthy and involved story than any of our previous projects. Oliver Wade, our animation director, compiled the scenes and found that we've got more than 60 minutes of movies. Even though most of them are about 30 seconds long, that's a lot of animation time. The problem was that we only gave our team of seven animators five months to animate them. That doesn't sound too bad until you consider that the animators creating the movies were also responsible for the in-game animations. Therefore they effectively had 2 and a half months. If you don't include weekends, that's about 10 seconds per animator per day. And that's a lot.

Fortunately, the animators had finished most of the in-game animations by the time the movies were in full swing. But it was still a real challenge. Furthermore, animating the scenes was just the first step. We had to add programmatic and 2D effects and convert many of the animations into MPEGs before alpha, which stretched many people to the limit.

We got such a late start because we had to finalize the story, write the scripts, audition the actors, record the dialogue, and put the final sound files together before starting the animation. It helped somewhat that we took an iterative approach — starting animations as soon as the first scenes were recorded — but in general the tardy start created a lot of stress.

3. Immense level designs. Even though we tempered our ambitions for the macro design, sometimes we cut loose and created some absolutely huge level designs. We had a habit of wanting to make each level better than the last, and a few times this tendency resulted in layouts that made the artists want to kill the designers.

Early on, we didn't have a good understanding of what "too big" meant. The first level designs we created were reasonable, but then we decided that we really needed to show off the power of the RATCHET technology. We also had some ambitious gameplay ideas involving a fight on a moving train and a hoverboard race. This resulted in the Metropolis and Blackwater City levels, two of the biggest in the game. When the artists saw the layouts they said, "Are you nuts? There's no way we can build this in six weeks!" So the designers went back to the designs and tried to edit them, but the levels still ended up being massive.

To the artists' and gameplay programmers' credit they made these and other huge levels work, and they did it on time. And to the designers' credit, they continued to find better and better ways to put more gameplay into smaller areas without sacrificing creativity. In the end, our level design ambitions pushed the limits of time and resources we had allotted.

Out of this stress came a more team-oriented approach to level design, where we now involve a large number of people — artists, programmers, sound engineers, and others — earlier in the design process. Whether or not levels in our future games will be smaller remains to be seen. But with more people involved at the beginning stages, we can find solutions sooner to balancing the need for gameplay space in levels with the time we have available to build them.

4. Maya issues. Maya is a superb tool for building polygonal environments and characters, and it's also great for animation and for prototyping particle effects, rendering, and many other things. However, early in the project we had decided to use Maya as our construction, texturing, lighting, and gameplay placement tool. We had abandoned our in-house tool, Karma, which we had used previously to do gameplay placement, texturing, and lighting. What we didn't realize was that with the size of our levels, we would push Maya past the breaking point.

Even though we set people up with dual 1.2GHz Dells with superfast graphics cards and a gigabyte of RAM, Maya would still chug and frequently crash

IMAGE: THE SNOWMAN © 2002 DUCK SOUP STUDIOS

AN INTERGALACTIC SHOWCASE OF SUPERIOR INTELLIGENCE IN COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES

CONFERENCE 27-31 JULY 2003 EXHIBITION 29-31 JULY 2003
SAN DIEGO CONVENTION CENTER SAN DIEGO, CALIFORNIA USA

www.siggraph.org/s2003

SIGGRAPH 2003
SAN DIEGO

SPONSORED BY ACM SIGGRAPH

whenever our levels got up to around 40MB. And forget making all 500K polygons in a level visible. Fortunately, Al Hastings and T.J. Bdelon worked valiantly to create a suite of plug-ins and tools that worked with the Maya API. This solution didn't always prevent the crashes that plagued the artists or the occasionally corrupted level, but it kept us running and allowed us to create finished levels every six weeks.

While Maya has always been and probably will still be our first choice for art creation, we're moving back to our original approach of using proprietary tools for things like gameplay setup, lighting, and texturing.

5. Localization woes. From the beginning we planned to include the NTSC and PAL versions of the game on one disc. This plan created two problems for us. First, we had to send all of our assets to Europe for localization in French, Italian, German, and Spanish as early as possible. In most cases this meant pre-alpha, which really put the squeeze on the animators who were working on the movies. Second, we knew that we would end up fixing both functionality and localization bugs at the same time. We anticipated that this would create even more chaos during the last few weeks before we went gold. And we were right.

Surprisingly, the biggest nightmare for us was the text localization. We had made the decision to allow subtitles for all of the movie scenes; plus we had a lot of text for the help system and a ton for the menus. We used spreadsheet databases to ensure some organization for all of the text (as opposed to entering localized text in the actual code, which we did on the SPYRO series), and this allowed us make updates and changes quickly. But the system was also prone to user error when cutting and pasting changes into the database.

Because we were still fixing TRC (technical requirement checklist) bugs — things like memory card messages — we were making text changes up to a couple of weeks before gold. We had also added some text late in the process to support some of our postgame features.

We made mistakes, and the localization folks in Europe made mistakes when putting fixes into the database. In addition, it took forever to transfer our discs to Europe once they were burned (eight hours to FTP if nothing crashed, 24 hours for a courier). These facts combined meant that we were still desperately trying to resolve some TRC issues hours before the gold disc was due. Fortunately, the game shipped on-time in all territories, but I think it prematurely aged our producer in Europe, as well as a few of us here.

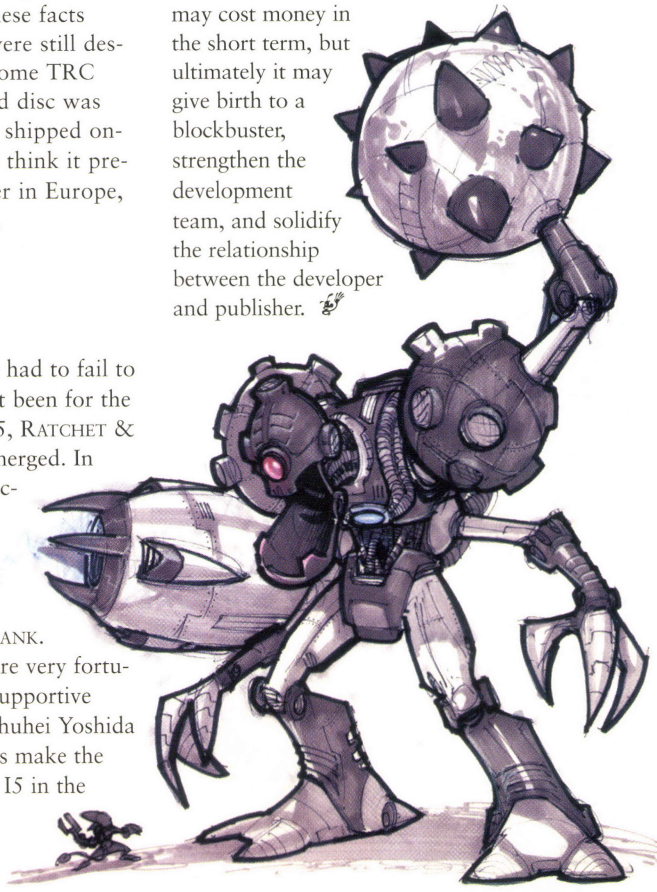
The Will to Kill

With this project, we had to fail to succeed. Had it not been for the pain we went through on I5, RATCHET & CLANK might have never emerged. In the six months of preproduction on I5 we learned how to make games on the PS2, and we were able to hit the ground running when we switched to RATCHET & CLANK.

Most importantly, we were very fortunate to have an extremely supportive publisher in Sony. SCEA's Shuhei Yoshida and Connie Booth helped us make the agonizing decision to shoot I5 in the head. But they made sure we understood that if we wanted to continue down that dark path of developing I5 for release, they would still support us.

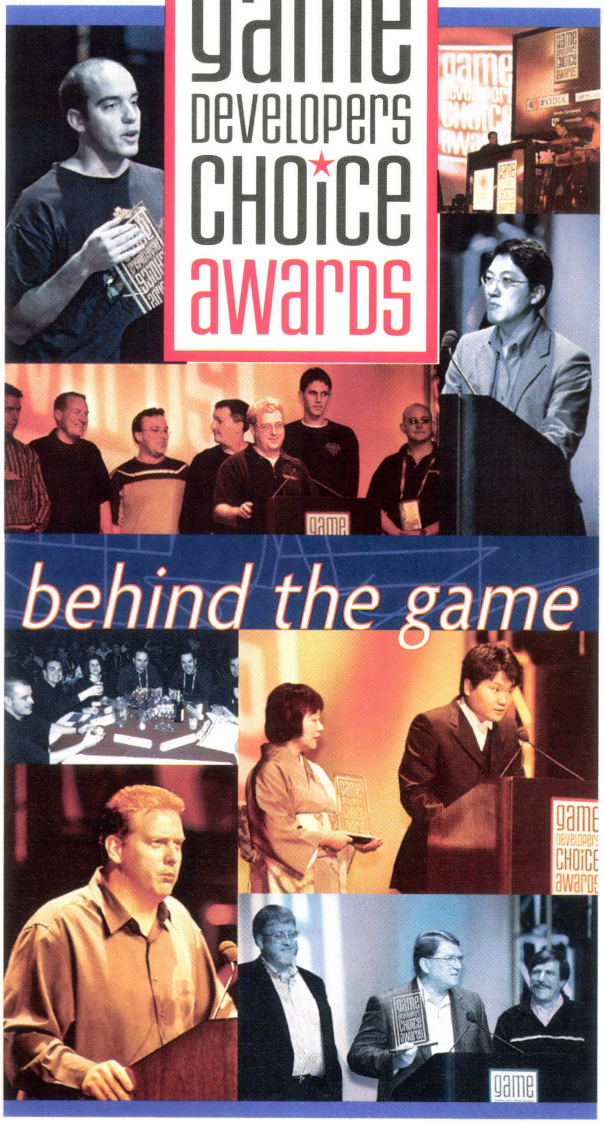
Furthermore, Sony never once threatened to cancel the I5 project or sever our relationship. Instead, they helped us to develop what Mark Cerny calls "the will to kill" — meaning we grew the balls to voluntarily throw out everything we had worked so hard on for six months and start over.

The development process that RATCHET & CLANK represents as a finished game is the ultimate example of how developer-publisher relationships can and should work. Sometimes good teams make games that aren't good. When a developer has the support of a great publisher and can cut off a nonperforming project in preproduction without fearing reprisals, everyone can save millions in production costs and apply the lessons learned to the next project. Doing so may cost money in the short term, but ultimately it may give birth to a blockbuster, strengthen the development team, and solidify the relationship between the developer and publisher.



The International Game Developers Association congratulates the recipients of the 3rd Annual

game developers choice awards



GAME OF THE YEAR
METROID PRIME
by Retro Studios

ORIGINAL GAMFE CHARACTER OF THE YEAR
Sly Cooper from
SLY COOPER AND
THE THIEVIUS RACCOONUS
by Sucker Punch Productions

ROOKIE STUDIO OF THE YEAR
Retro Studios for
METROID PRIME

EXCELLENCE IN AUDIO
Jack Grillo, Rebecca Hanck, Erik Kraber and Yuan Liu for sound effects in MEDAL OF HONOR: ALLIED ASSAULT

EXCELLENCE IN GAME DESIGN
Romain de Waubert de Genlis and Team for game design in BATTLEFIELD 1942

EXCELLENCE IN LEVEL DESIGN
Metroid Team for level design in METROID PRIME

EXCELLENCE IN PROGRAMMING
Mark Brockington, Scott Greig, Jason Knipe, Don Moar and Don Yakielashek for network programming in NEVERWINTER NIGHTS

EXCELLENCE IN VISUAL ARTS
Tetsuya Nomura for art direction in KINGDOM HEARTS

EXCELLENCE IN WRITING
Clint Hocking and JT Petty for writing in TOM CLANCY'S SPLINTER CELL

GAME INNOVATION SPOTLIGHTS
ANIMAL CROSSING
by Nintendo
BATTLEFIELD 1942
by Digital Illusions
MEDAL OF HONOR:
ALLIED ASSAULT by 2015, Inc.
THE THING
by Computer Artworks

LIFETIME ACHIEVEMENT AWARD
Gunpei Yokoi, creator, GameBoy

THE FIRST PENGUIN AWARD
David Crane, Larry Kaplan, Jim Levy, Alan Miller and Bob Whitehead, founders, Activision

IGDA AWARD FOR COMMUNITY CONTRIBUTION
Doug Church

Sponsored by: NVIDIA. RenderWare

Media Sponsors: G4 TV 4 GAMERS GAMESPOT

igda international game developers association

www.igda.org/awards

LUCASARTS ENTERTAINMENT COMPANY LLC

WELCOMES YOU TO

JOIN OUR FAMILY!

To Apply:

resumes@lucasarts.com/www.lucasarts.com
LucasArts Entertainment Company LLC
Attn: HR Recruiting
PO Box 10307
San Rafael, CA 94912

We Want You

Art
Animators, Level Artists, 3D Modelers

Design
Level Designers, Game Designers

Programming
3D Graphics, AI, PS2, Maya

Send resume and portfolio/reel to:

hiring@crankypg.com

Cranky Pants Games
803 Kirkland Avenue
Suite 200, Kirkland, WA
98033

Crankypants™ games
A THQ COMPANY
www.crankypantsgames.com

KONAMI
COMPUTER ENTERTAINMENT HAWAII

Frogger's Adventures 2
THE LOST WAND™

Hop to the challenge...

Konami offers a comprehensive benefits package and is searching for driven individuals with a passion for games.

We currently have positions open for:

- Lead Programmer
- Programmer
- Artists
- Sound Designer

Please contact us at HawaiiHR@konami.com or visit our website at:
www.konamihwi.com/jobs.php

Walter & COMPANY

Making a Critical Career or Business Decision?

BARBARA WALTER, CPC
CAREER COACH

760-745-4100

walterco@earthlink.net

http://www.sandiegomag.com/careers

FREE INITIAL CONSULTATION

Are you Ready for Life on the eDGE?

Edge of Reality, Ltd. is a veteran entertainment software developer focused on next generation consoles. Located in beautiful Austin TX, the studio is currently hiring for a major license based project. Edge of Reality is currently looking for talented individuals with a passion for games to join our highly creative team!

eDGE oF rEALITY

Positions Available

- Senior Environment Artist
- Senior Character Artist
- Environment Artist
- Senior Animator
- Senior Tools programmer
- Effects/Lighting artist
- Senior Texture Artist
- Lead Designer

See our Web site for more details! **www.edgeofreality.com**

tAKING gAMES tO tHE eDGE

CRYSTAL DYNAMICS

Crystal Dynamics, Inc., a subsidiary of Eidos Interactive, is looking for talented, creative candidates to join our multi project studio. If you think you have what it takes to be the next superstar in the game industry, check out our website for our current openings at www.eidosinteractive.com. Please send resume submissions to cooljobs@crystald.com and reference Gamasutra.

ORBITAL MEDIA

50524f4752414d4d4552532057414e544544

(PROGRAMMERS WANTED)

Developing & Publishing Games for the Game Boy Advance

www.orbitalmedia.com
Canada

ORBITAL MEDIA

WANTED
Digital Artist Designers

Developing & Publishing Games for the Game Boy Advance

www.orbitalmedia.com
Canada

GOT GAME? GOT GAME? GOT GAME?

INFOGRAMES

With hard-core, genre-defining games such as Driver™, Enter The Matrix™, Neverwinter Nights™, Stuntman™, Test Drive®, Unreal® Tournament 2003, and Unreal® Championship; and mass-market and children's games such as Backyard Sports™, Nickelodeon's Blue's Clues™ and Dora the Explorer™, Civilization®, Dragon Ball Z® and RollerCoaster Tycoon®, Infogrames has all kinds of excitement in store.

GOT GAME? GOT GAME?

Infogrames also offers:

- *Challenging and creative work environment*
- *Competitive salary and Bonus Plan*
- *401K plan with matching contributions*
- *Competitive Benefits Programs*
- *Paid time off*
- *An opportunity to work among the best minds and talent in the industry*

With our distinguished line-up of titles and projects, we are seeking the most talented people for positions that include:
Executive Producers, Level Designers, Senior Artists, Game Designers, Cinematic Artists, 3D, Game Programmers, Software Engineers
Your qualifications must include a smashing track record and determined passion, no excuses. **Slackers need NOT apply!**

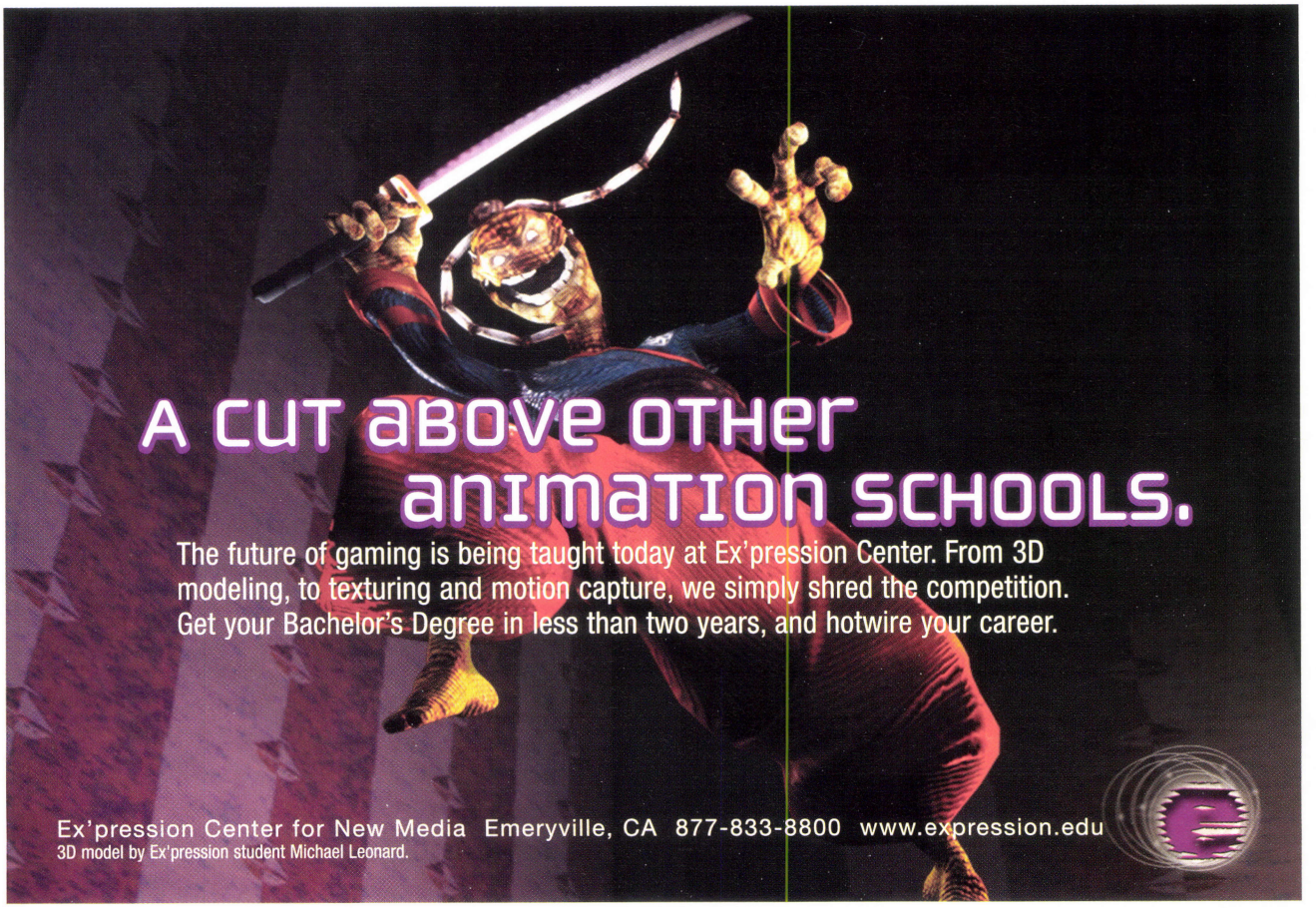
Send your resume and title history to **gamers@us.infogrames.com**

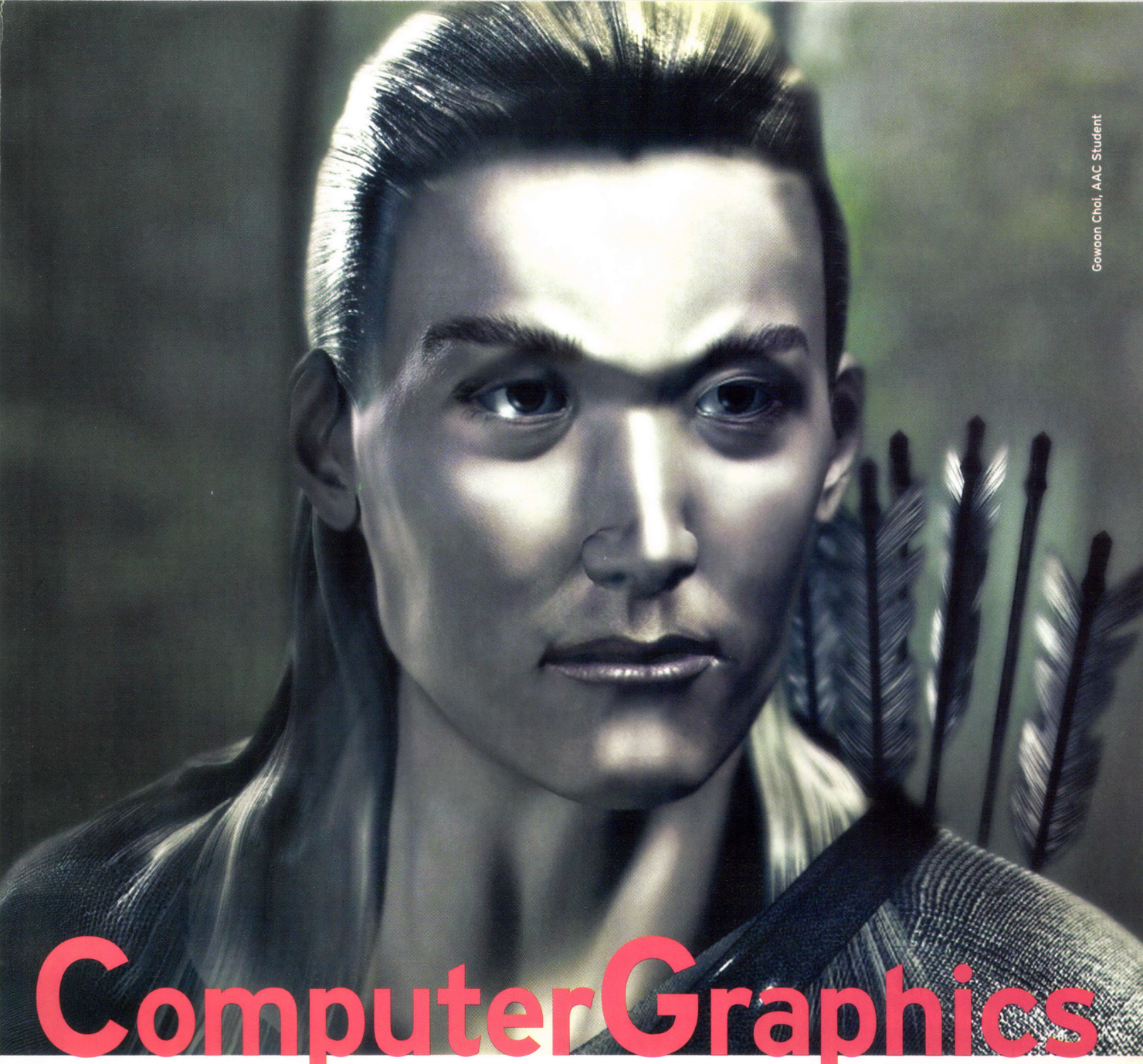
For more information about Infogrames, visit us on the Web at **www.infogrames.com.**

GOT GAME? GOT GAME?

www.us.infogrames.com

INFOGRAMES





ComputerGraphics

- 2D & 3D Animation
- 3D Modeling
- Character Design
- Computer Graphics
- Digital Imaging
- Filmmaking
- Game Design
- Visual Effects
- Web Design and more...

AA | BFA | MFA Degrees | Portfolio Development | Online Programs

Apply Now for Fall, Spring & Summer Semesters

High School Scholarships & Teacher Grants Available

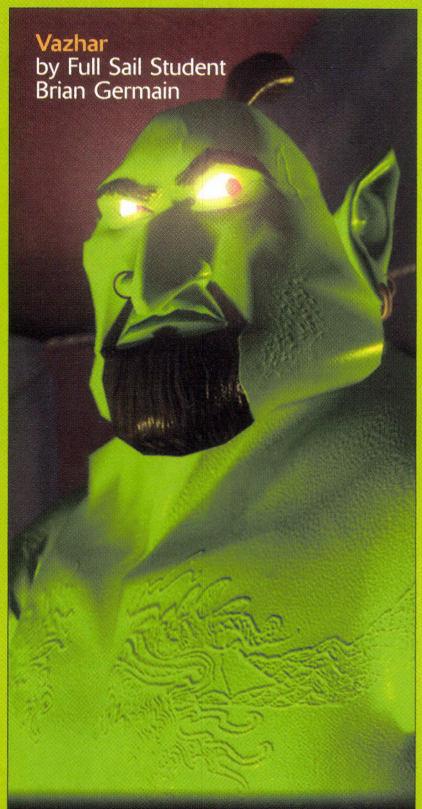
Hire Professional Graduates Today

AcademyOfArtCollege

1.800.544.ARTS | www.academyart.edu

79 New Montgomery St. | San Francisco, CA 94105 | Nationally Accredited by ACICS, NASAD & FIDER | **Established in 1929**

GETEDUCATED



FULL SAIL
Real World Education

School of:
Game Design
Computer Animation
Digital Media
Film
Audio
Show Production

800.226.7625
www.fullsail.com

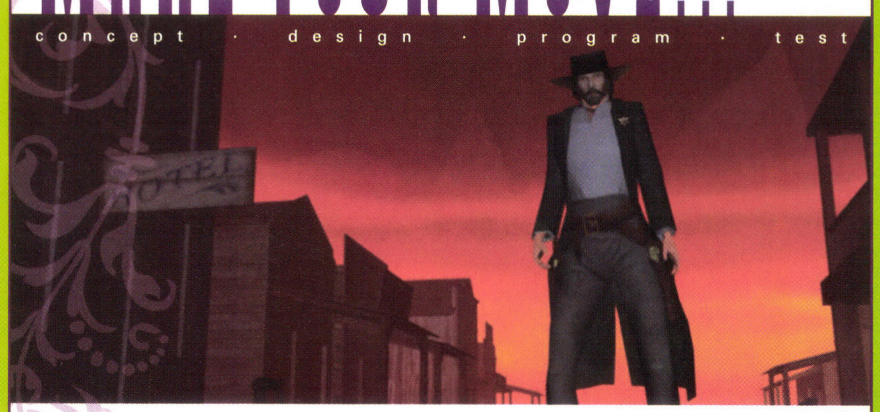
3300 University Boulevard
Winter Park, FL 32792

- Financial aid available to those who qualify.
- Job placement assistance.
- Accredited by ACCSCT.

©2001 Full Sail, Inc. All rights reserved.
The terms "Full Sail," "Full Sail Real World Education," and the Full Sail logo are either registered service marks or service marks of Full Sail, Inc.

MAKE YOUR MOVE...

concept · design · program · test



artwork: ryan munk

get a career in **game design**. get in the game @ **collins college**.

call today. **1.888.356.7777** 1140 s priest • tempe, arizona 85281 • www.collinscollege.edu

If you're going to pay for an orange...

you should get
an orange.

If you squeeze the new DVD from VFS you will never again be confused about who's got the juice.

To get more information or your own DVD of VFS student work call 1-800-661-4101 or email dvd@vfs.com

VFS Vancouver Film School
Creative. Disciplined. Focused.

Buglord
ALIENWARE
One Game. 3 Mobile Platforms.
Thousands of Angry Insects.
• 15 Single Player Levels for Palm, PocketPC and Windows
• Full Multiplayer Support
• Win prizes in our Multiplayer Tournament, including an Alienware Area 51m
www.buglord.com

watching your every move...
red eye studio
-motion capture for the masses-
1.847.843.2438
www.redeye-studio.com
RED EYE

do something with your life
or not.
See over there, where it looks like nothing is going on? That's your future if you just sit there.
> A **real** college degree, focused on advancing technology.
Available on campus or online, right where you're sitting.
University of Advancing Technology
UAT
Learn. Experience. Innovate.
Learn more.
www.uat.edu or 800.658.5744

...we hear it

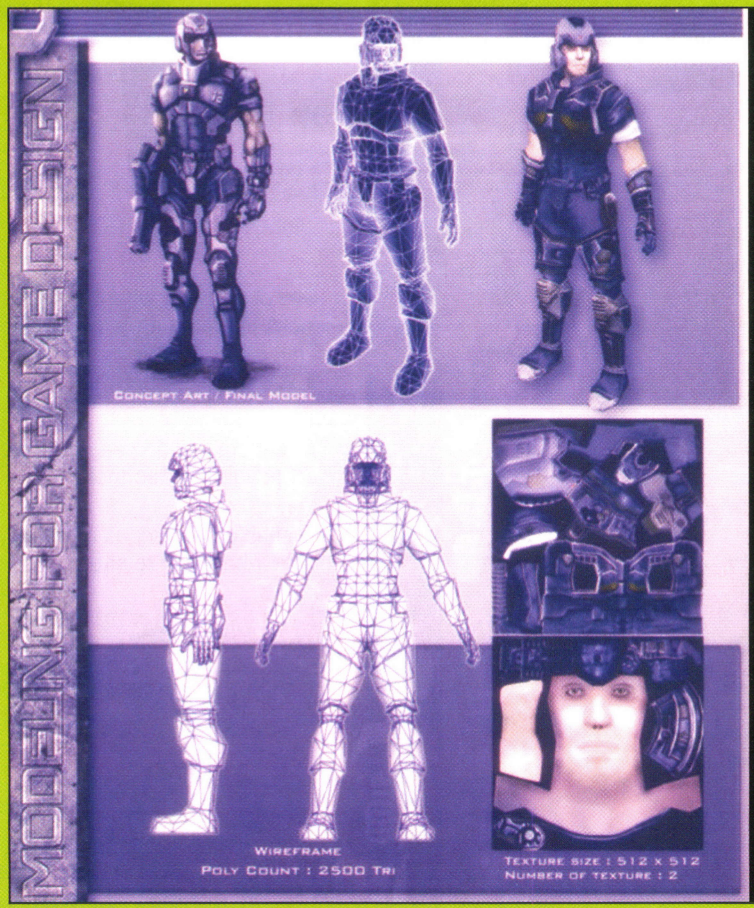
sound design
music production
voice talent

singularity-studios.com | info@singularity-studios.com

SingularityStudios

To target those looking to Get Educated, contact Aaron Murawski at 415-947-6227 or e-mail amurawski@cmp.com

To advertise in Creative Careers, contact Raelene Maiben at 415-947-6225 or e-mail rmaiben@cmp.com



Apply online at www.scad.edu

Animation | Architectural History | Architecture
Art History | **Broadcast Design** | Fashion | Fibers
Film and Television | Furniture Design | Graphic Design
Historic Preservation | Illustration | Industrial Design
Interactive Design and Game Development
Interior Design | Media and Performing Arts
Metals and Jewelry | Painting | Photography
Sequential Art | Sound Design | **Visual Effects**

B.F.A. | M.ARCH. | M.A. | M.F.A.

Savannah College of Art and Design

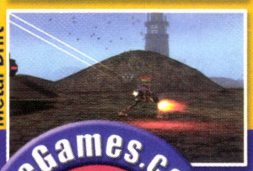
Savannah, Georgia | 800.869.7223

Thitipong Jitmakusol, Bangkok, Thailand
Character, Adobe Photoshop, Alias/Wavefront Maya

Invest in Your Own Game!

Realm Wars

Marble Blast


Metal Drift

- Award-Winning Network Code
- Engine and Tool Source
- Powerful Scripting Language
- Seamless Indoor & Outdoor Render Engine
- Supports PC, Mac and Linux Platforms

$100 per programmer seat for Independent developers. Commercial licenses available with NO royalties. The **Torque Game Engine** is supported and expanded by one of the world's largest communities of independent developers.

TORQUE GAME ENGINE

GarageGames.com • INDEPENDENT GAMES •

GarageGames

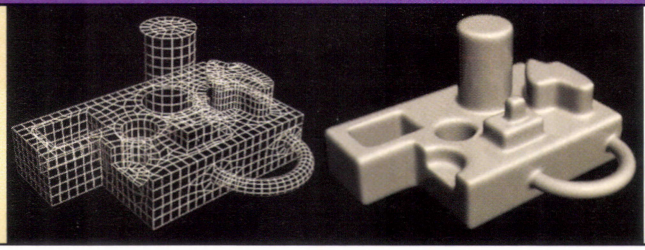
www.garagegames.com

345 Mill St., Suite #200 • Eugene, OR 97401 • 541.345.3040

©2002, GarageGames, Inc. Torque Game Engine, *Realm Wars* and GarageGames are trademarks of GarageGames, Inc. *Marble Blast* is a trademark of Monster Studios, Inc. *Metal Drift* is a trademark of Ozo Interactive.

Commercial licenses now available with NO royalties.

NPOWER SOFTWARE



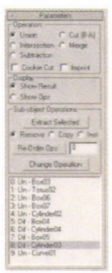
GD Conf. Booth # 1147

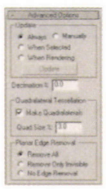
1) Power Booleans 2.0 - NEW:
   -- Quad Meshing for Sub-D
   -- Level Editing User Interface
2) Power Booleans Rhino - NEW
3) Power Solids on Rhino - NEW

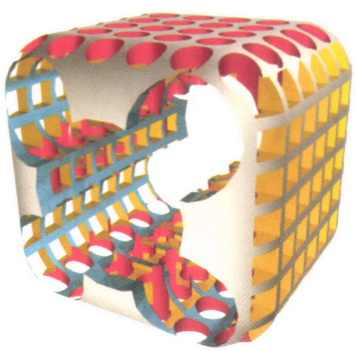
POWER BOOLEANS

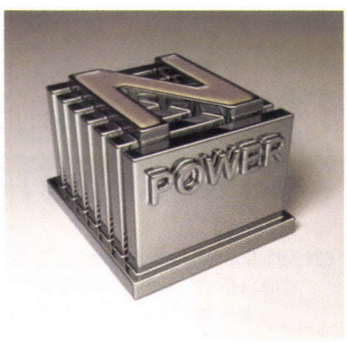
Power Booleans is a MAX Plug-in that provides extreamely reliable booleans on MAX meshes!

Fewer Polygons & Cleaner Meshes.
Fully Animatable Boolean Operations!
Improved Usability with Multi-Object Selection.
Integrated Decimation Polygon Reduction!









POWER SOLIDS

Power Solids introduces a traditional CAD Boundary representation into 3DSMAX!

View Dependent Tessellation for Rendering.
Import IGES, Rhino,STEP & SAT files.
Powerful Construction Primatives.
Filleting, Shelling & Booleans!
Animation of all Solids Operations.

www.npowersoftware.com

(858)-538-3800

TARGETPAVILION

To advertise in Target Pavilion, contact Aaron Murawski at 415-947-6227 or e-mail amurawski@cmp.com

To advertise in Creative Careers, contact Raelene Maiben at 415-947-6225 or e-mail rmaiben@cmp.com

...we make models
...we make textures
...we make animation

NEW PENCIL
WE MAKE ART

80 Liberty Ship Way, Suite 6 Sausalito, CA 94965
phone 415.339.1800 fax 415.339.1803
web www.newpencil.com

continued from page 72

and initiatives will pay off exponentially for the industry. This mix of research and coordinated outreach, combined with other initiatives, will burst open the dam.

So what does this mean for your better future as a game developer? It means possible new avenues of revenue and a stabilizing second market to steady things amid the unstable world of game development. No doubt there will eventually be games — built and funded for "serious" purposes — achieving crossover commercial success.

What should your role be? First, you should realize that this is a different business. Serious games require adjusted practices, which could include forgoing royalty upside, publicly releasing source code, creating features for instructors as well as players, entirely different funding means (including government contracting), and looking at projects that while potentially significant, may not result in a blockbuster payday. Also, project ramp-ups may be longer, and the clients may need some "What's a game and how does it work?" hand-holding. Many projects may be smaller in revenue, but there already exist project budgets with seven figures attached to them. So if you want to participate in this new arena, bring your skills, but adapt your methods.

Finally, Serious Games needs your help. If we are to bring the full force of the game industry into new areas, be they to create simulations for policy makers and the public or to show that e-learning can be truly fun, it has to be an organized vocal effort. You can help by getting involved with projects such as ours. We are announcing a number of initiatives via SeriousGames.org, the IGDA, and elsewhere, giving you ample opportunities to contribute, and likewise benefit. The rewards, besides new revenue, just might include helping to build a better world.

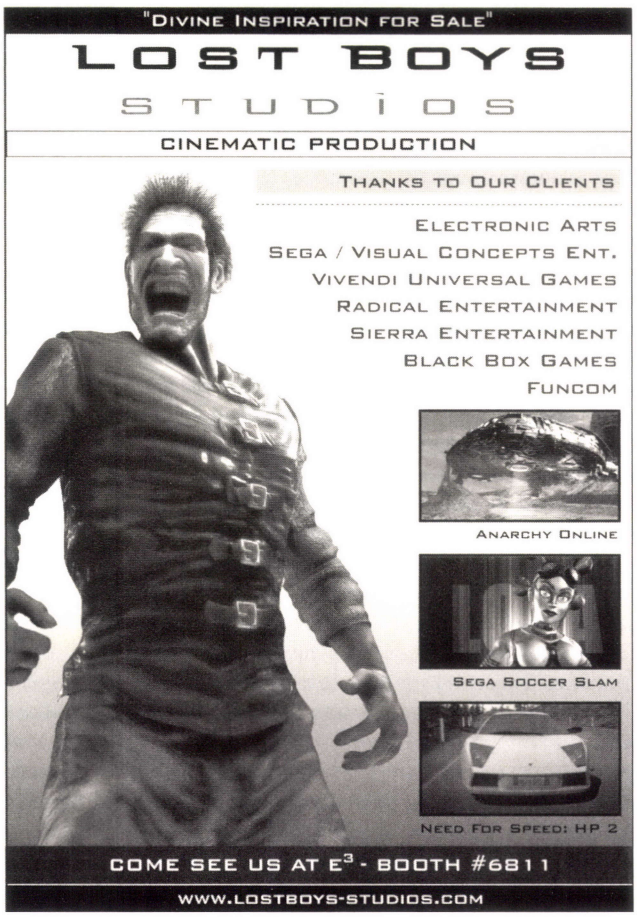
BEN SAWYER | *Ben is president of Portland, Maine–based Digitalmill, Inc. He assists several "serious game" projects and spearheads outreach efforts for the Woodrow Wilson Center's Serious Games Initiative. Contact him at bsawyer@dmill.com.*

FOR MORE INFORMATION

Serious Games Initiative: www.seriousgames.org
VIRTUAL U: www.virtual-u.org
IGDA: www.igda.com
MIT's Games-to-Teach: http://cms.mit.edu/games/education/

ADVERTISER INDEX

COMPANY NAME	PAGE
3Dlabs	33
Academy of Art College	65
AGI Media	C2
Alias|Wavefront	27
AMD	45
Anthro	15
Big World	7
Butterfly.net	29
Collins College	66
Cranky Pants Games	63
Criterion Software	2–3
Discreet	41
Edge of Reality	63
EMI Music Publishing	13
Expression Center for New Media	64
Full Sail Real World Education	66
Garage Games	69
Infogrames	64
Ingrooves	70
Integrity Ware	69
Lost Boys	71
Microsoft	47
New Pencil	70
Nokia	16
Numerical Design Ltd.	19–20
NXN	51
Octopi	67
Oregon 3D	48
Perforce	23
Programmer's Paradise	11
Qualcomm	C3
RAD Game Tools	C4
Red Eye Studios	67
Savannah College of Art & Design	68
Singularity Studios	68
Vancouver Film School	66



Gaming Our Way to Our Better Future

New Organizations and Pioneering Initiatives Are Creating New Outlets for Game Developers

You never know when serendipity will strike. One day you're trying to figure out what to do, the next moment the phone rings and you're on a plane to New York to meet with a major foundation. That was my situation two years ago, and now I'm in the middle of what could be a potentially explosive new outlet for game developers and publishers.

The result of that phone call and airline trip was VIRTUAL U, a college administration simulation developed by Trevor Chan and Enlight Software for the Alfred P. Sloan Foundation. Today that single game, in use at dozens of colleges, has pushed me onto the Serious Games Initiative at the Woodrow Wilson Center. Serious Games is establishing a series of projects leading up to the sustained creation of policy and management games for government and nongovernmental organizations. Dave Rejeski, leader of the Serious Games Initiative, calls it "gaming our way to a better future," a better future for the world in general. In VIRTUAL U's case, the better future is someone's college education. In a game developer's case, it's about a better future for the individual, the industry, and our society.

The idea behind both projects is that games, or in these cases, game-based simulations built by game developers, can be a compelling new generation of entertaining and effective experiential tools for people dealing with complex systems and systems management issues. The bigger scope is that games as a media form are serving to disseminate information, knowledge, and critical thinking. This epiphany isn't new to game fans or developers, but it's quietly becoming such to the world at large, its governments, and other major forces.

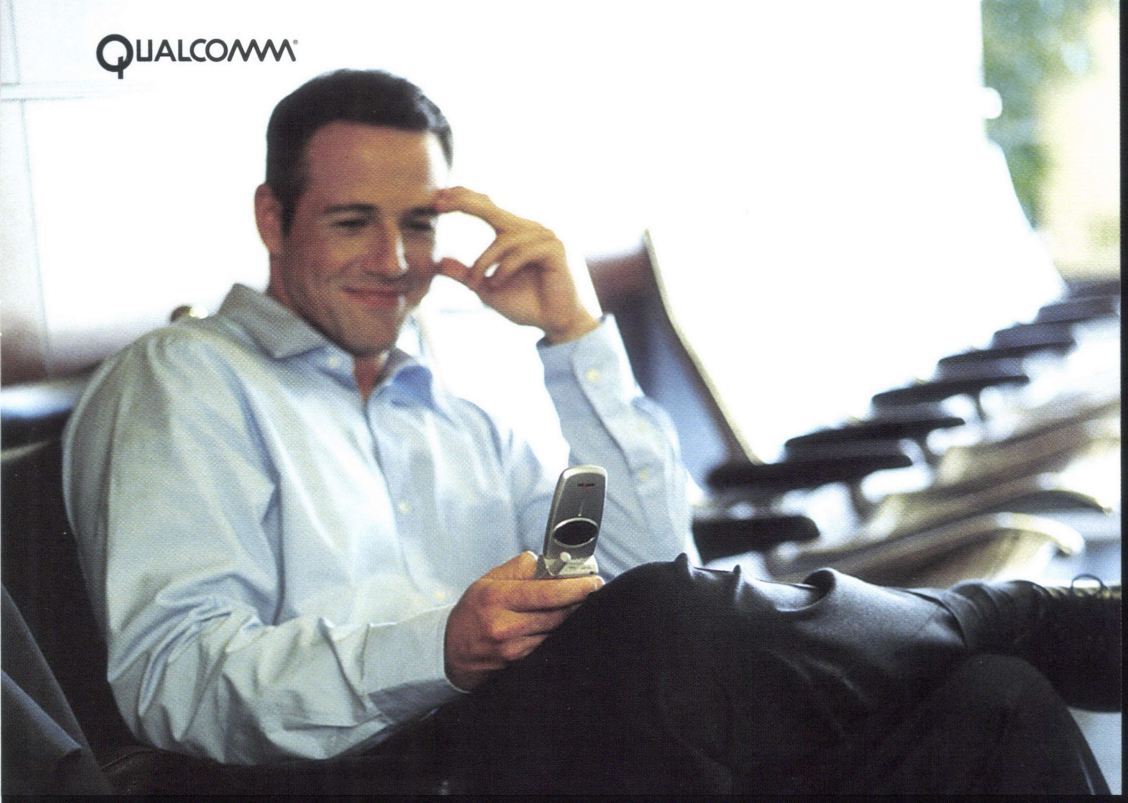
I am not proposing that until VIRTUAL U came along a game developer hadn't earned substantial revenues developing a game for non-entertainment purposes, nor would I suggest game skills haven't been applicable to other endeavors. However, what's happening now is not the sporadic or disconnected nature of things in the past or the second wave of READER RABBIT eduware. It's bigger than that; it's multi-faceted, organized, and aimed at new targets. As Rejeski points out, "Our government spends billions each year on simulation games alone." That's just one market, and now the military is getting serious about games (AMERICA'S ARMY is the tip of the iceberg). As others have evangelized, including *Digital Game-Based Learning* author Marc Prensky, the lucrative corporate market, once stung by poor e-learning offerings during the Internet bubble, is now waking up. To put some numbers behind the corporate opportunity, according to freelance game consultant (and *Game Developer* design columnist) Noah Falstein, Shell Oil spends close to a billion dollars a year in training and corporate learning. A supermarket chain I met with, once I told them a game might cost two million dollars, replied, "That's what we spend on toilet paper."

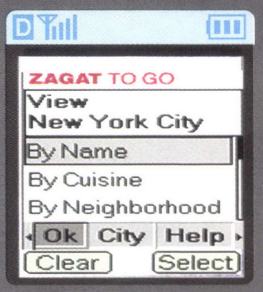
From previous hits or misses in years past, several threads and a growing history are earning emerging recognition. We are better able to preach new applicable uses of games. The Serious Games Initiative is indicative of this, and we are working to provide improved visibility for games to be seen as tools, including ways to see more projects in the mold of SIMHEALTH, VIRTUAL U, or AMERICA'S ARMY be created. Nor are we alone. M.I.T.'s Games-to-Teach project is organizing research on an entire matrix of demonstrated learning capabilities of games. Carnegie Mellon University's Entertainment Technology Center is becoming a hotbed of applied game-technology transfer. Both the International Game Developers Association and the Interactive Digital Software Association are also making contributions, and were present at a conference on this subject last February in Washington, D.C. They know a sustained flow of such products

continued on page 71

Illustration by Tyrone McCarthy

QUALCOMM





Navigation

Games

Ringers

MILLIONS OF POTENTIAL CUSTOMERS. ANY TAKERS?

It's called "Get It Now.℠" And it's the name Verizon Wireless, the nation's largest wireless carrier, has chosen for its new nationwide, BREW-enabled service. Dozens of BREW™ applications are now heading directly from the hands of developers into the hands of consumers with the help of the BREW Distribution System – and the financial rewards for these developers are returning just as fast. And as more carriers prepare to launch their BREW-enabled services, BREW applications such as games, email, news, weather, stock trades, position location and ringers are finding a potential market of literally millions upon millions of users. If you aren't developing for BREW, you aren't developing to your potential. To get started, visit www.qualcomm.com/brew/gd.

brew™
Customize. Personalize. Realize.™

©2003 QUALCOMM Incorporated. All rights reserved. QUALCOMM is a registered trademark of QUALCOMM Incorporated. BREW and Customize. Personalize. Realize. are trademarks of QUALCOMM Incorporated. Get It Now is a service mark of Verizon Wireless. JAMDAT and JAMDAT Mobile are trademarks or registered trademarks of JAMDAT Mobile, Inc. All rights reserved. ©2003 JAMDAT Mobile Inc. All other trademarks are the property of their respective owners.

THE BEST IN GAME DEVELOPMENT TECHNOLOGY

PIXOMATIC RENDERING TECHNOLOGY

Now you know what Michael Abrash and Mike Sartain have been up to! Designed for today's games, **Pixomatic** is their new software renderer that provides MIP mapping, bi-linear filtering, alpha-blending, alpha-test, point sprites, multi-texture, 32-bit and palettized textures, **DOT-3 bump mapping,** fog, specular, **stencil and z-buffering,** indexed primitives, multiple streams, high-speed blitting, and more! Pixomatic uses runtime code generation to create **optimized MMX, 3DNow!, and SSE** code on-the-fly! You will be amazed at its speed (what else would you expect from everyone's favorite optimizer?)! **Target the mass market** - don't let ancient 3D cards, buggy 3D drivers, or lack of 3D hardware prevent people from playing your games!

BINK VIDEO

BINK VIDEO

The very best video codec - make your videos shine! Unrivaled quality **(better than DVD** and MPEG II). Up to **three times faster** than other true-color codecs. Perceptibly lossless, **8 to 1 audio compression.** Decompresses to DirectDraw, DIBSections, YUV overlays, or any other memory. Available for Win32, MacOS, Xbox, and now **Nintendo GameCube.** Think Bink!

GRANNY 3D 2.2

Granny is a **sophisticated dynamic 3D animation system** with an optimized run-time engine that delivers incredible performance in a tiny memory footprint perfect for consoles, PCs and Macs. **MAX & Maya** plug-ins export materials, multiple vertex UVs and colors, mesh data, and complete animation and skinning information. Granny also now includes **amazing normal map generation** - give Granny a high and low-res model and she'll build the normal map! These features, plus advanced camera control, collision detection, animation and texture processing, MIP-map generation, and **Bink and S3TC** texture compression, all add years worth of features to your game in just a few hours of integration time!

MILES SOUND SYSTEM

The ultimate sound system for PC and Mac! Miles supports 2D and 3D audio, MIDI with DLS, streaming, CD audio, **DSP filtering, MP3 playback (patent rights included), Internet voice chat** (2900, 2400, and 1200 bps codecs), Creative **EAX 1, 2 & 3,** Aureal **A3D 1 & 2,** RSX 3D, DirectX 7, Dolby Surround, QSound, **super-fast software EAX emulation,** and more!

RAD GAME TOOLS ©

425.893.4300

www.radgametools.com

Powerful Technology. Easy To Use. No Royalties